Expressive Temporal Specifications for Reward Monitoring

IMPERIAL

Safe & Trusted AI
UKRI Centre for Doctoral Training

Omar Adalat Francesco Belardinelli
Imperial College London, UK

Motivation

- Designing **informative reward functions is hard**; **sparse rewards** can severely slow down learning in long-horizon tasks.
- **Temporal logic** offers a compact way to specify goals, but Boolean (true/false) semantics often still produce sparse feedback.
- **Key idea:** use **quantitative** temporal specifications $\mathrm{LTL}_f[\mathcal{F}]$ to produce a *dense, graded* reward signal that tracks progress towards satisfaction.

Quantitative temporal specifications $\mathrm{LTL}_f[\mathcal{F}]$

- Formulas are interpreted with values in $[0, 1]$ over a finite trace prefix $\lambda_{1:i} = (s_1, \ldots, s_i)$.
- Quantitative semantics preserves the temporal structure of LTL_f while allowing *degrees* of satisfaction.
- Quantitative atoms come from a state labelling function $\mathcal{L}$ (e.g., normalized distance-to-goal, energy remaining, safety margin).
- Under quantitative semantics, partial progress, e.g. *eventually(balanced)*, written as $F(\mathit{balanced})$, translates to $\max(\mathit{balanced})$, yielding intermediate rewards across timesteps.

Specifications using $\mathrm{LTL}_f[\mathcal{F}]$

- Can represent interesting temporal properties, such as Eventually F, Always G, and Until $\mathcal{U}$, using temporal and the usual propositional connectives (and $\wedge$, or $\vee$, negation , implication$\rightarrow$, etc.).

Example (Specification for the Cartpole environment)

Move the cart left or right to keep a pole balanced upright for as long as possible. Reward-specification pairs (for both the Boolean and quantitative setting):

$$\big((F(G(\mathit{reach\_goal})), 2),\ (G(\mathit{balanced}), 4)\big),$$

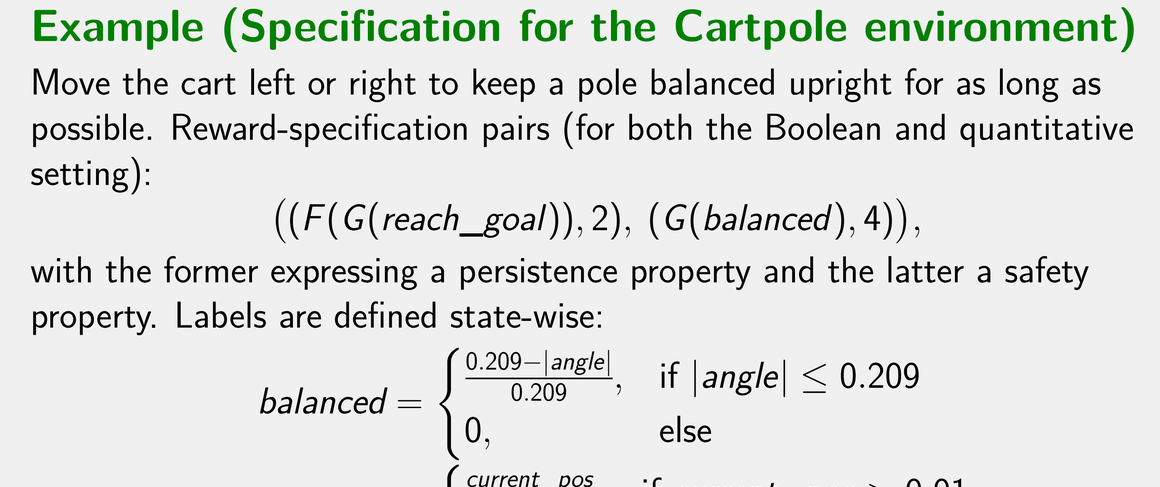
with the former expressing a persistence property and the latter a safety property. Labels are defined state-wise:

$$\mathit{balanced} = \begin{cases} \frac{0.209 - |\mathit{angle}|}{0.209}, & \text{if } |\mathit{angle}| \leq 0.209 \\ 0, & \text{else} \end{cases}$$

$$\mathit{reach\_goal} = \begin{cases} \frac{\mathit{current\_pos}}{\mathit{goal\_pos}}, & \text{if } \mathit{current\_pos} > 0.01 \\ 0, & \text{else} \end{cases}$$

Quantitative Reward Monitors (QRM)

- A QRM is a finite-state machine with real-valued *registers*.
- Registers track subformula values required by temporal operators (e.g., running min/max for G/F).
- The monitor outputs a reward each step: $r_i = \rho\, \mathcal{V}(t_{\mathit{reward}})$ for a specification-reward pair (φ, ρ).



(a) Boolean reward monitor (LTL_f). **(b)** Quantitative reward monitor ($\mathrm{LTL}_f[\mathcal{F}]$).

Figure: A boolean monitor (LTL_f, left) and quantitative monitor ($\mathrm{LTL}_f[\mathcal{F}]$, right) for $\neg a \mathcal{U}(a \wedge Fb)$.

Synthesis guarantees

- **Inductive construction** on the syntax of φ (with memoization of submonitors).
- **Correctness:** the reward register matches the quantitative semantics on every trace prefix.
- **Efficiency:** number of monitor states/transitions is linear in the size of the formula.

Learning with monitors

- Multiple specifications (φ_i, ρ_i) are combined into a single composite monitor $\mathcal{A}_c$.
- We learn on the synchronous product (extended) MDP:

$$\mathcal{M} \otimes \mathcal{A} \quad \text{with state space} \quad \mathcal{S}' = Q \times \mathcal{S}.$$

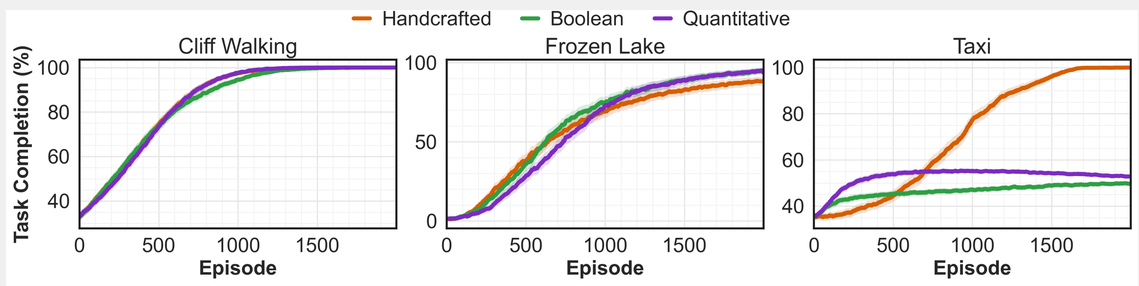
- Standard RL algorithms apply directly; a Markovian policy on $\mathcal{S}'$ suffices to realize the non-Markovian objectives.

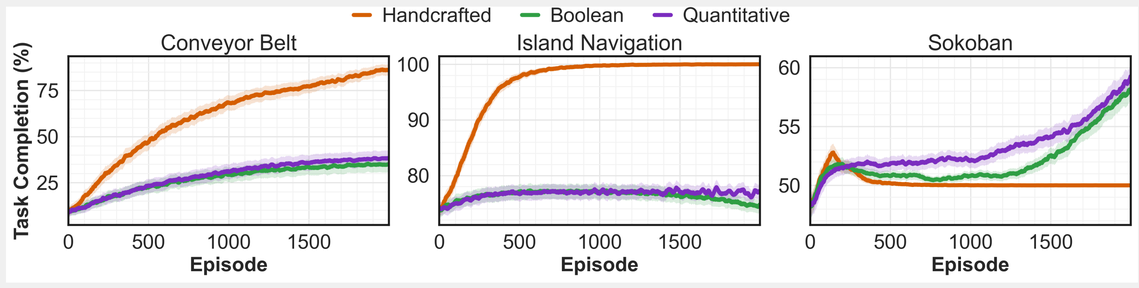
Safety specifications

- Safety goals require that once violated, rewards are *clamped* for the rest of the episode (no recovery to positive return).
- We syntactically identify safety formulas and treat them as veto monitors that can block rewards from other monitors after a violation.

Empirical evaluation

- Benchmarked on Gymnasium environments spanning Classic Control, Toy problems, Box2D, and Safety Gridworlds.
- Compared: environment reward (Gymnasium), Boolean monitor LTL_f, and quantitative monitor $\mathrm{LTL}_f[\mathcal{F}]$.
- Metrics: time/episodes to convergence and a quantitative task-completion score.





Takeaways & outlook

- Quantitative monitors provide dense rewards directly from high-level temporal specifications and incorporate safety constraints.
- Performance depends on the quality of quantitative fluents; when informative, QRMs can reduce convergence time and improve task completion.
- Future work: richer quantitative connectives (e.g. "soon", "within X timesteps") and extensions in the direction of the multi-agent setting.

The research described in this paper was partially supported by the EPSRC (grant number EP/X015823/1) and the UKRI Centre for Doctoral Training in Safe and Trusted AI (grant number EP/S0233356/1).