# Reward Machine Construction Writeup

Maximilian Stollmayer

## Basic Definitions

**Definition.** Propositional symbols *are statements that are either true or false.* Formulas *over propositional symbols consist of combinations of them with operations* $\neg$, $\wedge$, $\vee$, $\implies$ *and* $\iff$ *. We say a formula* $\psi$ provable *from a formula* $\varphi$*, if* $\psi$ *can be derived from* $\varphi$ *and write* $\varphi \vdash \psi$*.*

In the following we will suppose that we are in a reinforcement learning setting with a finite set of states $S$, with $s_0$ being the initial state, a set $T \subseteq S$ of terminal states and a finite set of actions $A$. Furthermore we suppose that we have a finite set of propositional symbols $\mathcal{P}$.

**Definition.** *A* labeling function $L : S \times A \times S \to 2^{\mathcal{P}}$ *maps experiences to truth assigments over propositional symbols* $\mathcal{P}$*.*

**Definition.** *A* Non-Markovian Reward Decision Process (NMRDP) *is a tuple* $(S, A, s_0, T, R, \gamma)$*, where* $S, A, s_0, T$ *and* $\gamma$ *are defined as in a regular MDP and* $R : (S \times A)^+ \times S \to \mathbb{R}$ *is a non-Markovian reward function that maps finite state-action histories into a real value. Note that* $X^+ := \bigcup_{n=1}^{\infty} X^n$ *represents all non-empty finite sequences of a set* $X$*.*

## Reward Machines

**Definition.** *A* Mealy machine *is a tuple* $(U, u_0, \Sigma, \mathcal{R}, \delta, \rho)$*, where*

- $U$ *is a finite set of states*
- $u_0 \in U$ *is the initial state*
- $\Sigma$ *is a finite input alphabet*
- $\mathcal{R}$ *is a finite output alphabet*
- $\delta : U \times \Sigma \to U$ *is the transition function*
- $\rho : U \times \Sigma \to \mathcal{R}$ *is the output function*

**Definition.** *A* reward machine (RM) *is a Mealy machine* $(U, u_0, \Sigma = 2^{\mathcal{P}}, \mathcal{R}, \delta, \rho)$*, where* $\mathcal{R}$ *is a finite set of reward functions* $S \times A \times S \to \mathbb{R}$.

**Definition.** *The non-Markovian reward function* $R$ induced by an RM $(U, u_0, 2^{\mathcal{P}}, \mathcal{R}, \delta, \rho)$ *is*

$$R : (S \times A)^+ \times S \to \mathbb{R}$$
$$(s_0, a_0), \ldots, (s_n, a_n), s_{n+1} \mapsto \rho\big(u_n, L(s_n, a_n, s_{n+1})\big)(s_n, a_n, s_{n+1})$$
$$R\big((s_0, a_0), \ldots, (s_n, a_n), s_{n+1}\big) = r(s_n, a_n, s_{n+1})$$

*where* $u_n = \delta\big(u_{n-1}, L(s_{n-1}, a_{n-1}, s_n)\big)$ *is defined recursively with the base case being the initial state* $u_0$.

# Logics and Automata

ltl & co, dfa, dfa construction theorem and proof (source?)

**Definition.** *A* deterministic finite automaton (DFA) *is a tuple* $(U, u_0, \Sigma, \delta, F)$*, where*

- $U$ *is a finite set of states*
- $u_0 \in U$ *is the initial state*
- $\Sigma$ *is a finite input alphabet*
- $\delta : U \times \Sigma \to U$ *is the transition function*
- $F \subseteq U$ *is a set of accepting states*

# Reward Specifications

**Definition.** *A* reward specification *is a set* $R = \{(r_1, \varphi_1), \ldots, (r_N, \varphi_N)\}$*, where each* $r_i \in \mathbb{R}$ *and* $\varphi_i$ *is a formula over the propositional symbols* $\mathcal{P}$ *expressed in some regular language.*

**Definition.** *Let* $\tau = \big((s_0, a_0), \ldots, (s_n, a_n), s_{n+1}\big) \in (S \times A)^+ \times S$ *be a trace of experiences. We say that the projection of the experiences of* $\tau$ *by* $L$ entails *a formula* $\varphi$*, and write* $\tau \vdash_L \varphi$*, if* $L(s_0, a_0, s_1) \ldots L(s_n, a_n, s_{n+1}) \vdash \varphi$.

**Definition.** *The non-Markovian reward function* $\hat{R}$ induced by the reward specification $R = \{(r_1, \varphi_1), \ldots, (r_n, \varphi_n)\}$ *assigns reward* $\hat{R}(\tau) := \sum_{k=1}^{N} \mathbb{1}(\tau \vdash_L \varphi_k)$ *to a trace* $\tau = \big((s_0, a_0), \ldots, (s_n, a_n), s_{n+1}\big) \in (S \times A)^+ \times S$.

## Construction Theorem

**Theorem.**
*There exists a reward machine that induces the same non-Markovian reward function as a given reward specification $R = \{(r_1, \varphi_1), \ldots, (r_N, \varphi_N)\}$.*

**Proof.**

Let...

**Corollary.**

## Example

*go through the construction of a very simple formula to reward machine*