Action Grammars: A Grammar Induction-Based Method for Learning Temporally Extended Actions

Robert Tjarko Lange [1] **Aldo Faisal** [2]

Abstract

Temporal abstraction allows efficient re-usability of sequential behavior across the state space. Learning semantically meaningful macro-actions defines a key challenge of Hierarchical Reinforcement Learning. Here, we introduce a fully end-to-end algorithmic framework which builds a grammatical memory buffer. By treating an agent's trace as a sentence sampled from the policy-conditioned environment, the agents learns hierarchical sub-structures using powerful unsupervised grammatical inference algorithms.

1. Introduction

Goal-driven behavior is inherently driven by the hierarchical nature of the imposed task. Hierarchical Reinforcement Learning (HRL) intends to solve the credit assignment problem of the agent by modeling different time-scales of decision making.

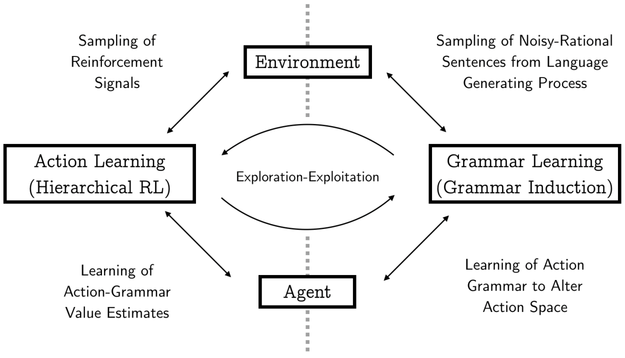

Figure 1. Action Grammars for HRL

2. Related Work

*Equal contribution [1]Einstein Center for Neurosciences Berlin, Berlin, Germany [2]Department of Computing, Imperial College London, London, United Kingdom. Correspondence to: Robert Tjarko Lange <robert.lange17@ic.ac.uk>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

3. Technical Background

3.1. Temporally-Extended Actions

Semi-Markov Decision Processes extend the classical MDP setting to incorporate not only environmental uncertainty but also time uncertainty. Instead of dealing with a Dirac waiting distribution, the time between individual decisions is modeled as a random variable, $\tau \in \mathbb{Z}_{++}$. It is described by the probability distribution $P(s', \tau | s, m)$ which characterizes the the joint likelihood of transitioning from state $s \in \mathcal{S}$ into state s' in τ time steps given action m was pursued.

Thereby, SMDPs allow one to elegantly model the execution of actions which extend over multiple time-steps (e.g. sequences of primitive actions or sub-policy execution). Multiple different hierarchical action structures have been proposed. In this work we focus on the most simplest, namely macro-actions. Simply put, macro-actions specify the sequential and deterministic execution of multiple primitive actions.

3.2. Context-Free Grammars

4. Context-Free Action Grammars

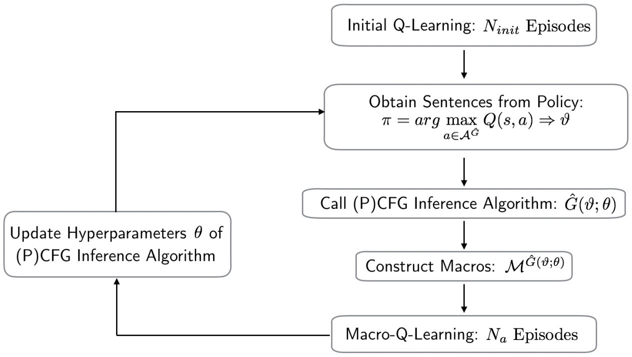
Initial Q-Learning: N_{init} Episodes
Obtain Sentences from Policy: $\pi = arg \max_{a \in \mathcal{A}^{\hat{G}}} Q(s,a) \Rightarrow \vartheta$
Call (P)CFG Inference Algorithm: $\hat{G}(\vartheta; \theta)$
Construct Macros: $\mathcal{M}^{\hat{G}(\vartheta;\theta)}$
Macro-Q-Learning: N_a Episodes
Update Hyperparameters θ of (P)CFG Inference Algorithm

Figure 2. Action Grammars: A General Pipeline

$$L(\theta) := \mathbb{E}_{s,m,r^{\tau_m},s',\tau \sim D_{\tau_m}} [(r^{\tau_m} + \gamma^{\tau_m} \max_{m'} Q(s', m'; \theta^-) - Q(s, m; \theta))^2]$$

5. Experiments

5.1. Towers of Hanoi

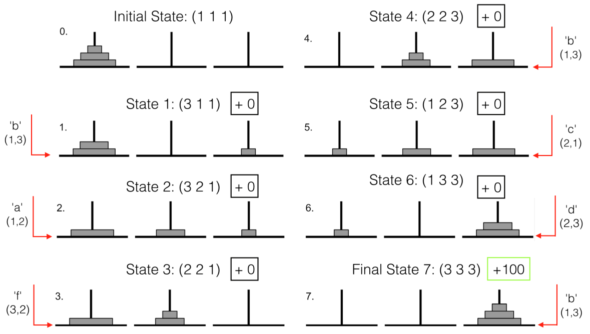

Figure 3. Optimal Solution for Towers of Hanoi Environment for 3 disks ($N = 3$)

Visualize the grammar learned

5.2. ATARI Environments

6. Discussion

This work has introduced a computational framework which connects the field of context-free grammar inference with Hierarchical Reinforcement Learning. Going forwards, we interested in extending this approach to both options as well as probabilistic grammars. One potential way of achieving this might be by defining a probabilistic termination condition with the help of syntactic surprise(**??**). Furthermore, building a grammatical dictionary might be promising for efforts in continual or life-long learning.

Algorithm 1 Bubble Sort

```
Input: data $x_i$, size $m$
repeat
  Initialize $noChange = true$.
  for $i = 1$ to $m - 1$ do
    if $x_i > x_{i+1}$ then
      Swap $x_i$ and $x_{i+1}$
      $noChange = false$
    end if
  end for
until $noChange$ is $true$
```