# Action Grammars: A Cognitive Model for Learning Temporal Abstractions

**Robert Tjarko Lange (robert.lange17@imperial.ac.uk)**
Einstein Center for Neurosciences Berlin, Chariteplatz 1, 10117, Berlin, Germany

**Aldo Faisal (a.faisal@imperial.ac.uk)**
Imperial College London, South Kensington Campus, London SW7 2AZ, United Kingdom

## Abstract

**Hierarchical Reinforcement Learning algorithms have successfully been applied to temporal credit assignment problems with sparse reward signals. However, state-of-the-art algorithms require manual specification of sub-task structures, a sample inefficient exploration phase and lack semantic interpretability. Human infants, on the other hand, efficiently detect hierarchical sub-structures induced by their surroundings. In this work we propose a cognitive-inspired Reinforcement Learning architecture which uses grammar induction to identify sub-goal policies. More specifically, by treating an on-policy trajectory as a sentence sampled from the policy-conditioned language of the environment, we identify hierarchical constituents with the help of unsupervised grammatical inference. The resulting set of temporal abstractions is called *action grammars* (Pastra & Aloimonos, 2012) and can be used to enable efficient imitation, transfer and online learning.**

**Keywords:** Decision Making; Reinforcement Learning; Computational Linguistics

## Introduction

Genetically inherited inductive biases enable human infants to infer hierarchical rule-based structures from language, visual input as as well as auditory stimuli (M. C. Frank, Slemmer, Marcus, & Johnson, 2009; Marcus, Fernandes, & Johnson, 2007). Several MEG and fMRI studies provide evidence for a universal process of hierarchical language comprehension in the brain (S. L. Frank & Christiansen, 2018; Brennan, Stabler, Van Wagenen, Luh, & Hale, 2016; Nelson et al., 2017) that extends to motor control (Pastra & Aloimonos, 2012; Stout, Chaminade, Thomik, Apel, & Faisal, 2018). By processing trajectories of an expert, the infant is able to learn policies over higher level sequences of low level control elements. Inspired by these observations, this work proposes to overcome the problem of sub-structure discovery in Hierarchical Reinforcement Learning (HRL) by making use of grammatical inference. More specifically, the HRL agent uses grammar induction to extract hierarchical constituents from trajectory sentences. The proposed solution to the credit assignment problem is split into two alternating stages (see fig. 1):

1. **Grammar Learning**: Given episodic trajectories we treat the time-series of transitions as a sentence sampled from the language of the policy-conditioned environment. Using grammar induction algorithms (Nevill-Manning & Witten, 1997) the agent extracts hierarchical constituents of the current policy. Based on the estimated production rules, temporally-extended actions are constructed which convey goal-driven syntactic meaning. The grammar can efficiently be inferred (linear time) and provides enhanced interpretability.

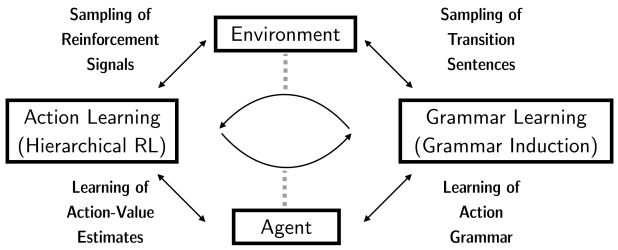



Figure 1: Action grammars closed alternation loop.

2. **Action Learning**: Using the grammar-augmented action space, the agent acquires new value information by sampling reinforcement signals form the environment. They refine their action-value estimates using Semi-Markov Decision Process (SMDP) Q-Learning (Bradtke & Duff, 1995). By operating at multiple time scales, the HRL agent is able to overcome difficulties in exploration and value information propagation. After action learning, the agent samples simulated sentences by rolling out transitions from the improved policy.

By alternating between stages of grammar and action value learning the agent iteratively reflects and improves on their behavior in semi-supervised manner. The inferred grammar parse trees are easy to interpret and provide semantically meaningful sub-policies. Our experiments highlight the effectiveness of the action grammars framework for imitation, curriculum and transfer learning given an expert policy rollout. Furthermore, we show promising results for an online version which iteratively refines grammar and value estimates.

## Background

**Temporal Abstractions.** SMDPs extend Markov Decision Processes to account for not only reward and transition uncertainty but also time uncertainty. The time between individual decisions is modeled as a random variable, $\tau \in \mathbb{Z}_{++}$. The waiting time is characterized by the joint likelihood of transitioning from state $s \in \mathcal{S}$ to state $s'$ in $\tau$ time steps given action $m$ was pursued, $P(s', \tau | s, m)$. Thereby, SMDPs allow one to elegantly model the execution of actions which extend over multiple time-steps. A macro-action (McGovern,

Sutton, & Fagg, 1997), $m \in \mathcal{M}$ specifies the sequential and deterministic execution of multiple ($\tau_m$) primitive actions. Let $r_{\tau_m} = \sum_{i=1}^{\tau_m} \gamma^{i-1} r_{t+i}$ denote the accumulated and discounted reward for executing a macro. Value estimates can then be updated using SMDP-Q-Learning (Parr, 1998) in a model-free bootstrapping-based manner:

$$Q(s,m)_{k+1} = (1-\alpha)Q(s,m)_k + \alpha\left(r_{\tau_m} + \gamma^{\tau_m} \max_{m' \in \mathcal{M}} Q(s',m')_k\right)$$

The DQN (Mnih et al., 2015) objective can then be adapted to the semi-Markov case:

$$L(\theta) := \mathbb{E}[(r_{\tau_m} + \gamma^{\tau_m} \max_{m' \in \mathcal{A} \cup \mathcal{M}} Q(s',m';\theta^-) - Q(s,m;\theta))^2]$$

The gradient with respect to the parameters is approximated by Monte Carlo samples from the experience replay (Lin (1992); ER) buffer $\{s,m,r_{\tau_m},s',\tau_m\} \sim D_{\mathcal{M}}$. The learning dynamics can be stabilized by making use of a target network and gradient clipping.

**Context-Free Grammars.** Given a start symbol $S$, a formal grammar $(\Sigma, \mathbb{N}, S, \mathcal{P})$ produces an output of strings. Production rules $\mathcal{P}$ map a set of non-terminal vocabulary $\mathbb{N}$ either to another non-terminal or terminal string within the terminal vocabulary $\Sigma$. Context-free grammars (CFG) (Chomsky, 1959) constrain the set of productions to either map from one-to-one, one-to-none or one-to-many. A non-branching and loop-free CFG is called a straight-line grammar. Given a sample of sentences, grammar induction infers a consistent language grammar. Sequitur (Nevill-Manning & Witten, 1997) sequentially reads in all symbols and collects repeating subsequences of symbols into a production rule. The final encoded string is only allowed to have unique bigrams and inferred production rules must be used more than once in the derivation of the string. In order to overcome Sequitur's problem of noise overfitting, $k$-Sequitur (Stout et al., 2018) has been proposed. Instead of replacing a bigram with a rule if the bigram occurs twice, it has to occur at least $k$ times. As $k$ increases the grammar becomes less prone to overfitting and the resulting grammar is more parsimonious in terms of production rules.

## Context-Free Action Grammars

Just like communication, action sequences convey goal-directed semantic meaning. They consist of hierarchical structures and are conditioned on the environment in which they are uttered. Furthermore, many real world problems require a hierarchy of subgoal achievements which increase in sequential difficulty and timescale. A trajectory obtained from traversing the current policy $\pi$ can be viewed as a sample from the language generated by the policy-specific grammar, $L(\pi|E)$. Let the terminal vocabulary $\Sigma$ consist of the primitive action space $\mathcal{A}$, hence $\Sigma = \mathcal{A}$. We denote $\vartheta^i \sim L(\pi|E)$ for $i = 1,\dots N_g$ trajectories. Given a set of trajectories, a CFG estimate $\hat{G}$ can be inferred and the resulting production rules transformed into macro-actions $\mathcal{M}^{\hat{G}}$ by recursively flattening the non-terminals. The action space of the agent is then augmented such that $\mathcal{A}^{\hat{G}} = \mathcal{A} \cup \mathcal{M}^{\hat{G}}$. Depending on the generating policy of the compressed traces, we propose several grammar-based HRL agents.

**Expert & Transfer Grammars.** If the traces $\vartheta^i$ are sampled from the language $L(\pi^\star|E)$ generated by the optimal policy, the agent can use the resulting grammar macros in an imitation learning setting. Before the onset of the first value learning stage, the action space is augmented with the optimal productions. Furthermore, an agent faced with learning a curriculum of tasks can make use of the optimal grammar of an easier solved task. Skills universal to all tasks do not have to be re-learned at every stage. Instead, the inferred optimal grammar provides an effective knowledge structure which accelerates the agents learning process.

**Online Inferred Grammars.** If an episode successfully terminated, the grammar inference process identifies repeating sub-goal achieving patterns. We hypothesize that by extracting action grammar sub-sequences, one compresses the temporal dimension of the credit assignment problem. After each grammar compression step, the action space is augmented with a new set of grammar macros. The previous set becomes inactive. In order to preserve value estimates between updates, we propose three solutions: (1) *Transfer learning* (Oquab, Bottou, Laptev, and Sivic (2014), see fig. 2): To accommodate the variable set of grammar-inferred skills, the size of the DQN output layer has to be updated. Transferring the value-relevant feature detectors between action space augmentation, allows the agent to use the previously learned value characteristics. (2) *Grammar ER Buffer*: It is necessary to maintain a grammar-altered buffer system in order to store transition tuples specific to previously inferred macro-actions. At any given point the agent can only sample macro transitions which are associated with the currently active set of grammar macros. Thereby, sample efficiency is increased once a grammar macro is repeatedly inferred. (3) *Intra-Macro Updates*: During the execution of a macro-action, one stores the overall macro transition tuple $< s_t, m_t, r_{t+\tau_m}, s_{t+\tau_m+1}, \tau_m, "on" >$ as well as the individual transitions $\{< s_i, a_i, r_i, s_{i+1}, 1, "on" >\}_{i=t}^{t+\tau_m}$. Thereby the agent is able to exploit all gathered transition experiences throughout the overall learning process.

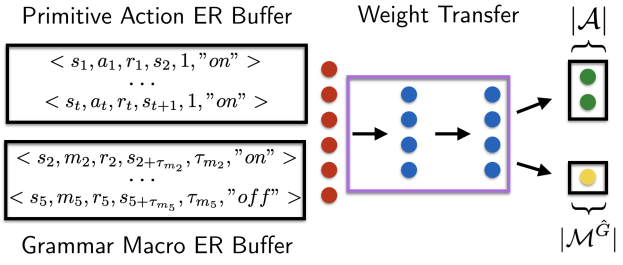



Figure 2: **Left.** Grammar Experience Replay Buffer **Right.** Grammar-DQN with adaptive output head.

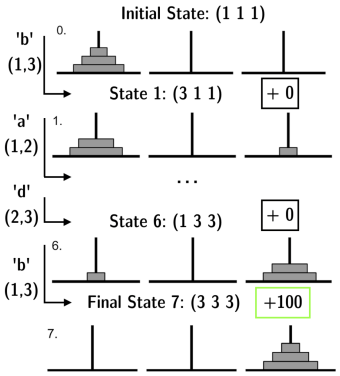
The length of the sampled trace is going to increase or de-

crease over the course of the learning procedure. The regularization parameter of the $k$-Sequitur grammar inference algorithm has to be adapted accordingly.

## Experiments

The goal of the following experiments is to answer the following questions: (1) Does a grammar learned from optimal policy rollouts allow for rapid imitation learning? (2) Can CFG grammars be used in order to enhance curriculum learning by the means of transferring previously learned action grammars? (3) Is online grammar inference and action space adaptation able to structure the exploration process of the HRL agent? In order to answer these question we choose the general $N$-disk Towers of Hanoi (ToH) environment (see fig. 3) as well as a hierarchically structured gridworld task (see fig. 4).

Solving the $N$-disk ToH problem requires the agent to identify a hierarchical and recursive principle. By moving $n-1$ disks onto an auxiliary pole and the $n$-th disk onto the target pole, the agent is able solve the sparse reward problem. Since such a routine can easily be formulated within a grammar parse tree, we hypothesize that the action grammars framework might provide an efficient solution.




Trace ($\vartheta$): bafbcdbafec
fbafbcdbcfecdbafbcdb

$\Sigma = \{a,b,c,d,e,f\}$

| | 2-Sequitur | |
|---|---|---|
| $\vartheta^{enc}$ | BCDfBEfDdBb | |
| $\frac{\lvert\vartheta^{enc}\rvert}{\lvert\vartheta\rvert}$ | 0.355 | |
| $\mathbb{N}$ | $\mathcal{P}$ | $\mathcal{M}^{2-Seq}$ |
| B | CEd | bafbcd |
| C | - | baf |
| D | - | ec |
| E | - | bc |

Figure 3: **Left.** Sparse Reward RL Formulation of the ToH Problem. **Right.** 2-Sequitur ToH (5 disks) Grammar-Macros.

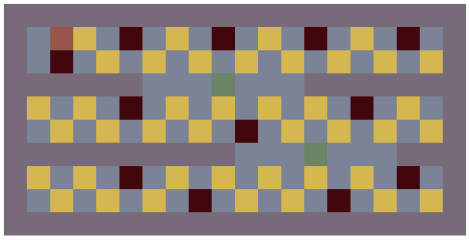


Figure 4: Hierarchically-Structured Grid World Environment.

The gridworld, on the other hand, provides a non-sparse reward design. The agent (red) has to avoid poisonous items (black) and collect food (yellow). Hence, the agent is required to solve a large set of individually smaller subtasks. Finally, the agent has to avoid a terminal collision with the moving blocks (green), whereas the ToH environment rewards the fastest solution.

**Learning with Expert & Transfer Grammars.** The right-hand side of figure 3 shows the grammar and resulting macros inferred from a trace of the optimal policy 5-disk ToH problem using the 2-Sequitur. The flattened production rule $B \rightarrow CEd \rightarrow bafbcd$ captures the recursive nature learned by the grammar. $C \rightarrow baf$ moves two disks on the auxiliary pole, while $E \rightarrow bc$ moves a third disk from source to target pole and one disk back onto the source pole. The Expert Grammar HRL agent's action space is augmented as follows:

$$\mathcal{A}^{\hat{G}} = \mathcal{A} \cup \mathcal{M}^{2-Seq} = \mathcal{A} \cup \{bafbcd, baf, ec, bc\}$$

Figure 5 displays learning results for different SMDP-Q-Learning agents with macro-actions defined by the production rules inferred from a single trace of the optimal policy.

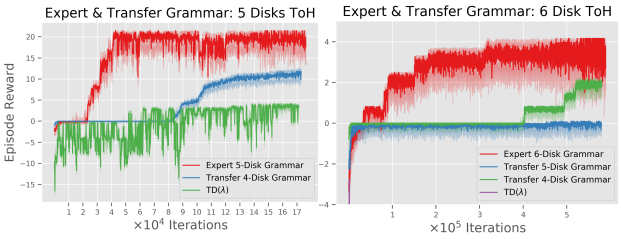



Figure 5: Expert & Transfer Grammar (*ToH*): **Left.** 5 Disk Environment. **Right.** 6 Disk Environment. Averaged over 5 random seeds. Median, 10th and 90th percentile.

The grammar macros accelerate the learning progress and reduce the variance of policy rollouts. We hypothesize that this is due to the temporal compression of the sequential problem provided by the macro grammars. Finally, the Transfer Grammar agent is capable of transferring the knowledge distilled in a simpler optimal grammar(4 disks) to a more complex setting (6 disks).

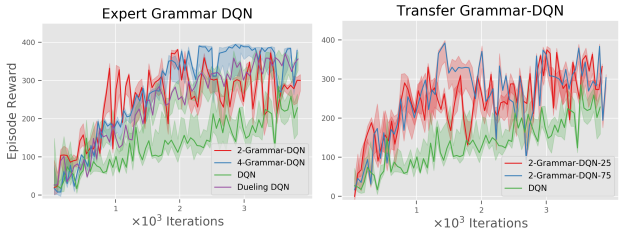



Figure 6: Expert & Transfer Grammar (*Gridworld*): **Left.** Expert Grammar. **Right.** Transfer Grammar. Averaged over 5 random seeds. Median, 10th and 90th percentile.

The gridworld Grammar-DQN agent (see fig. 6) again infers a set of macro-actions from a single expert rollout. Afterwards, the output layer and action space are augmented. The fixed architecture of the DQN is a two-layer 128 hidden units multi-layer perceptron trained using Adam (Kingma & Ba, 2014) with

a batch-size of 32. The two Expert Grammar-DQN agents differ in the amount of macro-actions (top two and four most used productions in the encoded policy trace) inferred with 2-Sequitur on a converged DQN agent rollout. Again, the expert grammar-endorsed agent is significantly accelerated in their initial learning progress. The two Transfer Grammar-DQN agents, on the other hand, infer a set of two grammar macros from a single sub-optimal separate DQN agent's (trained for 25 or 75 episodes) policy rollout. Our experiments show, that even with noisy non-optimal rollouts the grammar agents are able to exploit the inferred structure of the environment.

**Learning with Online Inferred Grammars.** Figure 7 displays the results of the online grammar inference framework for the gridworld task. Every 500 optimization steps the DQN agent infers a new set of grammar macros from a self-rollout using 2-Sequitur. We augment the action space with the top two most used flattened production rules in the trace compression. The learning dynamics provide a competitive extension to the general DQN framework.

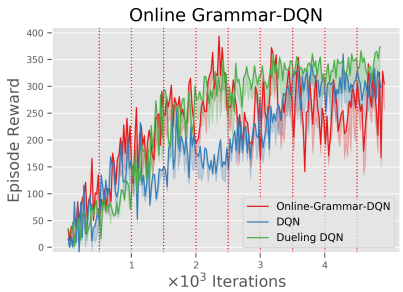



Figure 7: Online Grammar: Gridworld Grammar DQN. Averaged over 5 random seeds. Median, 10th and 90th percentile.

We want to emphasize the relationship between grammar inference and exploration. In our experiments we found that the frequency of grammar updating as well as the grammar inference hyperparameters play a crucial role.

## Conclusion

Inspired by hierarchical parse trees of sequential behavior, we introduced a novel cognitive decision making framework which exploits grammatical inference to identify temporally-extended actions. Our contributions are the following: (1) + (2) CFG-based HRL agents provide efficient and interpretable solutions to imitation and transfer learning tasks. (3) Alternating between grammar updates and learning action values is an effective method to learn an optimal grammar as well as an optimal policy online.

In future work we are interested in exploring stochastic grammars as well as their incorporation into model-based RL approaches. Ultimately, we envision a dictionary of action sequences which provides an expandable library of skills for agents which act in diverse naturalistic environments. This could provide a mayor contribution to a key endeavor in general artificial intelligence: Life-long learning.

## References


Bradtke, S. J., & Duff, M. O. (1995). Reinforcement learning methods for continuous-time markov decision problems. In *Advances in neural information processing systems* (pp. 393–400).

Brennan, J. R., Stabler, E. P., Van Wagenen, S. E., Luh, W.-M., & Hale, J. T. (2016). Abstract linguistic structure correlates with temporal activity during naturalistic comprehension. *Brain and language*, *157*, 81–94.

Chomsky, N. (1959). A note on phrase structure grammars. *Information and control*, *2*(4), 393–395.

Frank, M. C., Slemmer, J. A., Marcus, G. F., & Johnson, S. P. (2009). Information from multiple modalities helps 5-month-olds learn abstract rules. *Developmental science*, *12*(4), 504–509.

Frank, S. L., & Christiansen, M. H. (2018). Hierarchical and sequential processing of language. *Language, Cognition and Neuroscience*, *0*(0), 1-6.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, *8*(3-4), 293–321.

Marcus, G. F., Fernandes, K. J., & Johnson, S. P. (2007). Infant rule learning facilitated by speech. *Psychological science*, *18*(5), 387–391.

McGovern, A., Sutton, R. S., & Fagg, A. H. (1997). Roles of macro-actions in accelerating reinforcement learning. In *Grace hopper celebration of women in computing* (Vol. 1317).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540).

Nelson, M. J., El Karoui, I., Giber, K., Yang, X., Cohen, L., Koopman, H., ... others (2017). Neurophysiological dynamics of phrase-structure building during sentence processing. *Proceedings of the National Academy of Sciences*, 201701590.

Nevill-Manning, C. G., & Witten, I. H. (1997). Identifying hierarchical structure in sequences: A linear-time algorithm. *CoRR*.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1717–1724).

Parr, R. E. (1998). *Hierarchical control and learning for markov decision processes*. University of California, Berkeley Berkeley, CA.

Pastra, K., & Aloimonos, Y. (2012). The minimalist grammar of action. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *367*(1585), 103–117.

Stout, D., Chaminade, T., Thomik, A., Apel, J., & Faisal, A. A. (2018). Grammars of action in human behavior and evolution. *bioRxiv*, 281543.