

---

# Action Grammars: Grammar Induction-Based Learning of Temporal Abstractions

---

Robert T. Lange <sup>\*†</sup>

Einstein Center for Neurosciences Berlin  
Technical University Berlin  
robert.lange17@imperial.ac.uk

Aldo Faisal

Imperial College London  
Department of Computing & Bioengineering  
a.faisal@imperial.ac.uk

## Abstract

Hierarchical Reinforcement Learning algorithms have been successfully applied to large-scale problems with sparse reward signals. By operating at multiple time scales, the Reinforcement Learning agent is able to overcome difficulties in exploration and value information propagation. However, current approaches face one of three unsatisfying properties: They either require manual specification of hierarchical structures, lack clear interpretability or can hardly be justified in a comparative fashion. This work combats all of the shortcomings in a fully automated and end-to-end fashion. By treating an on-policy trajectory as a sentence sampled from the policy-conditioned language of the environment, we are able to apply powerful ideas from computational linguistics to the sub-structure discovery problem. We identify hierarchical constituents with the help of unsupervised grammatical inference. The resulting set of temporal abstractions is called *action grammar* and we show how to efficiently deploy such in multiple challenging Reinforcement Learning settings.

## 1 Introduction

Learning a policy over temporally-extended actions allows the Hierarchical Reinforcement Learning (HRL) agent to combat the uncertainty induced by single time-step decision making. The agent overcomes exploration problems, by restricting their decision process in a syntactically meaningful way. The biggest challenge of HRL is the actual identification of a meaningful substructure specification. As of yet, this challenge has not been successfully addressed. Current approaches require the optimization of many hyperparameters (e.g. the number of desired options, network architecture) and make strong assumptions regarding the initiation set of the subtask (e.g. the complete state space). This can easily lead to misspecification and results in a significant slow-down of learning and exploration. While providing a fully end-to-end approach, "deep" attempts lack interpretability and often times require severe amounts of pre-training.

Human infants, on the other hand, learn seemingly unstructured patterns in nature and from observing role models. They are incredibly well equipped to infer hierarchical rule-based structures from language, visual input as well as auditory stimuli [19, 13, 18]. By observing an expert, they get a head-start in their learning process and are able to learn over higher level sequences of low level control elements. Furthermore, there is convincing evidence from several MEG and fMRI studies that indicates a form of hierarchical language comprehension in the brain [10, 14, 4, 23] and a parallelism to motor control [27, 33]. Inspired by such observations this work overcomes the identified weaknesses by merging HRL with the field of computational linguistics. More specifically, we propose the usage of grammatical inference algorithms to extract hierarchical structures from

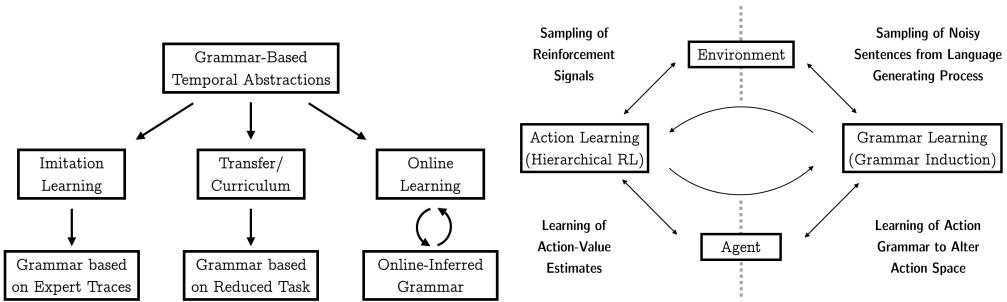
---

<sup>\*</sup>This work was done while R.T.L. was a Master's student in the FaisalLab & Imperial College London.

<sup>†</sup>Github repository: <https://github.com/RobertTLange/action-grammars-hrl>.

trajectory sentences with the ultimate aim to deploy them in the HRL process. Thereby, the original RL problem is split into two stages:

1. **Grammar Learning:** Given episodic trajectories we treat the time-series as a sentence sampled from the language of the policy-conditioned environment. The language in turn was generated by the grammar induced by the current policy. Using grammar induction the agent extracts hierarchical constituents of the current policy. Based on this estimate they constructs temporally-extended actions which convey hierarchical syntactic meaning. Afterwards, the agent's action space is augmented with such actions.
2. **Action Learning:** Using the grammar-augmented action space, the agent acquires new value information from interacting with the environment and refines his action-value estimates using Semi-Markov-Decision-Process-Q-Learning [3]. Afterwards, we sample simulated "sentences" from the improved policy by rolling out transitions in the environment.



**Figure 1:** Action Grammars for HRL. **Left.** Applicability of expert action grammars for imitation, transfer and online learning. **Right.** Online-inferred action grammars alternation loop. The procedure consists of alternating updates of the grammar estimate and a refinement of the corresponding "action grammar" value estimates

In the following we outline how a general grammar-augmented Reinforcement Learning procedure provides semantically meaningful solutions to key problems in RL (see left part of figure 1). We proceed in the following manner: First, we summarize the current state of sub-structure discovery for HRL and review the required technical background. Afterwards, we introduce our proposed framework and outline how grammar-based temporal abstractions are able to provide efficient, effective and interpretable sub-structures. Our experiments highlight the usability of the action grammars framework for imitation learning and transfer learning given an expert grammar. Furthermore, we display strong results of an online version which iteratively refines grammar and value estimates.

## 2 Related Work

We are not the first to infer hierarchical structure in subgoal achievement problems. More specifically, the option discovery problem deals with the question of how to construct a set of options [34] that captures the hierarchical structure between sub-regions of the core MDP. Roughly speaking the current state-of-the-art approaches can be categorized in three main pillars: First, graph theoretic [15, 22, 17, 28] and visitation-based [20, 32, 28] approaches aim to identify bottlenecks. Bottlenecks are regions in the state space which characterize successful trajectories. This work, on the other hand, identifies patterns solely in the action space and does not rely on rewardless exploration of the state space. Gradient-based approaches, on the other hand, discover parametrized temporally-extended actions by iteratively optimizing an objective function such as the estimated expected value of the log likelihood with respect to the latent variables in a probabilistic setting [8] or simply the expected cumulative reward in a policy gradient context [1, 31]. Grammar induction infers patterns without supervision solely based on a compression objective. The resulting parse tree provides an interpretable structure. Finally, multi-layer [2, 35, 12] approaches attempt to split the goal declaration and goal achievement across different stages and layers of the learning architecture. Usually, the top level of the hierarchy specifies goals in the environment while the lower levels have to achieve such. Such architectures lack sample efficiency and easy interpretation. Our context-free grammar-based approach, on the other hand, requires few rollout traces and can generalize to more difficult task-settings.

### 3 Technical Background

**Temporally-Extended Actions.** Semi-Markov Decision Processes (SMDP) extend the classical Markov Decision Process 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 joint likelihood of transitioning from state  $s \in \mathcal{S}$  into state  $s'$  in  $\tau$  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 [21, 34, 25, 9]. In this work we focus on the most simplest, namely macro-actions. Simply put, a macro-action,  $m \in \mathcal{M}$  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 [26] in a model-free manner:

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

In order to increase sample efficiency, it is recommended to perform intra-macro updates for each state transition tuple  $\{ < s, a, r, s' > \}_{\tau_m}$  within the macro execution. The experience replay buffer and DQN objective can also easily be refined to the case of temporally-extended actions:

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

**Context-Free Grammars.** Formal grammars and the theory of computational linguistics study both generating and accepting systems that underlie a language. Given a start symbol  $S$ , a formal grammar  $(\Sigma, \mathcal{N}, S, \mathcal{P})$  produces an output which is a string of words. The terminal vocabulary  $\Sigma$  is a set of terminal elements used to construct the sentences of a language.  $\mathcal{N}$  denotes the non-terminal vocabulary which is a set of elements only used in the process of deriving a sentence. The production rules  $\mathcal{P}$  are ordered pairs of strings such that  $\alpha \rightarrow \beta, \alpha \in V^+, \beta \in V^*$ . A type-2 grammar [6, 7], also known as context-free grammar (CFG) is such that the production rules have the following form:

$$A \rightarrow \beta, \text{ where } \beta \neq \lambda \text{ or } |\beta| \neq 0$$

Since production rules either map from one-to-one, one-to-none or one-to-many they are called context-free. The context of a non-terminal symbol does not influence the production rule [27]. A context-free grammar that is non-branching and loop-free is called a straight-line grammar [30]. Such grammars are restrictive since they are only capable of generating a single sentence.

The process of inferring a grammar for a language that is consistent with a given sample of sentences is called grammatical inference or grammar induction [16]. The smallest grammar problem [5, 30] formalizes the problem of finding the smallest CFG which compresses a string generated by a straight-line grammar. This problem turns out to be NP-hard [5]. Two greedy approximations to the smallest CFG are provided by Sequitur [24] and G-Lexis [29]. Given a single sentence of the language, Sequitur sequentially reads in all symbols and collects repeating subsequences of symbols into a production rule. Therewhile, the final encoded string is only allowed to have unique bigrams (*Digram Uniqueness*, [24]) and production rules must be used more than once in the derivation of the string (*Rule Uniqueness*, [24]). In order to overcome Sequitur's problem of noise overfitting,  $k$ -Sequitur [33] 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 discovered CFG grammar becomes less and less sensitive to overfitting noise and the resulting grammar is more parsimonious in terms of productions. Lexis [29] provides an optimization-based alternative which iteratively constructs a directed acyclic graph (DAG), the so-called Lexis-DAG. Starting from a trivial graph which connects a set of target sentences with the set of elements in the terminal vocabulary, the Lexis-DAG is constructed by adding intermediate nodes. The indirect objective is to minimize a cost function (e.g. number of concatenations or DAG edges) while imposing that the constructed graph satisfies a set of Lexis-DAG properties. Again, this problem by itself is NP-hard. G-Lexis, the greedy algorithmic implementation, searches for substrings that will lead to a maximal reduction in the cost, when added as new intermediate node.

## 4 Context-Free Action Grammars

Action sequences as well as communication by the means of words both convey meaning and are goal-directed. Both consist of hierarchical structures and are conditioned by the environment in which they are uttered in. The crucial assumption that connects linguistics with eager behavior is as follows:

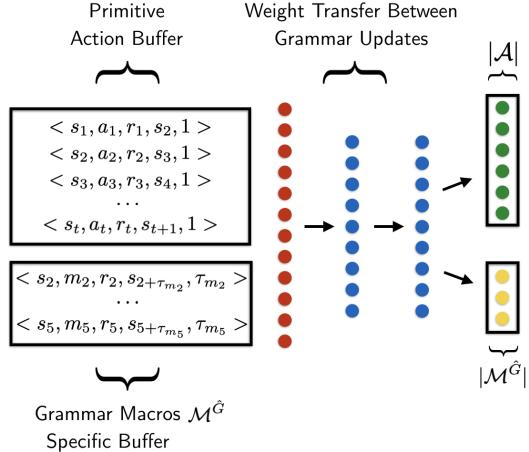
**Assumption 1.** *Observed episodic behavior (with trajectory  $\vartheta = \{\vartheta_1, \dots, \vartheta_T\}$  where  $\vartheta_t = \{s_t, a_t\}$ ) can be equivalently viewed as sentences sampled from the language,  $L(G)$  with  $G \sim \pi|E$ .*

Let us assume that the optimal policy of a Reinforcement Learning agent is hierarchically structured (i.e. with repeating sequences of actions) for a specific environment  $E$ . The optimal policy  $\pi^*$  then consists of a hierarchy of subgoal achievements which increase in sequential difficulty when moving up the hierarchy. We define the terminal vocabulary  $\Sigma$  to consist of the primitive action space  $\mathcal{A}$ , hence  $\Sigma = \mathcal{A}$ . A trajectory obtained from traversing the current policy  $\pi$  is viewed as a sample from the language generated by the grammar  $L(\pi|E)$ . We write  $\vartheta^i \sim L(\pi|E)$  for  $i = 1, \dots, N_g$  trajectories. Each trajectory has an individual length denoted by  $T_i$ . Given a set of trajectories,  $\vartheta^1, \dots, \vartheta^{N_g}$ , a context-free grammar estimate  $\hat{G}$  can be inferred. The resulting production rules can be transformed into macro-actions by recursive flattening. We augment the action space of the HRL agent, e.g.  $\mathcal{A}^{\hat{G}} = \mathcal{A} \cup \mathcal{M}^{\hat{G}}$ . Depending on the nature of the grammar compressed traces, the HRL agent can utilize the resulting semantically-augmented action space in different ways:

**Learning with Expert Grammar and Transfer Grammar.** If the traces  $\vartheta^i$  are sampled from the language  $L(\pi^*|E)$  generated by the optimal policy, the agent can use the resulting grammar and flattened macros in an imitation learning setting. Furthermore, an agent faced with learning a sequence of tasks with increasing difficulty can make use of the optimal grammar for the previous task. Thereby, skills which are universal to all tasks do not have to be re-learned at every tasks. Instead, the relevant sequential behavior is already captured by the grammar macro.

### Learning with Online Inferred Grammars

**Grammars.** The previous approach relied on a single set of static expert traces from which the agent once infers a grammar and then continues to perform SMDP-Q-Learning. Another approach is to alternate between a stage of grammar learning based on self-rollouts and a stage of action value refinements (see right part of figure 1). If the episode successfully terminated, the grammar inference process identifies repeating patterns that led to successful goal achieving experiences. By extracting these patterns and redefining them as temporally-extended actions, we can additionally *reflect and store* the progress made not only in the value estimate but also in the augmented action space. A natural question then becomes how to transfer values between two stages of value learning? In order to approximately preserve value estimates, we make use of transfer learning. More specifically, the classical DQN architecture can be altered as follows (see figure 2): In order to accommodate the variable set of grammar-inferred skills the size of the DQN output layer has to be updated after every grammar learning step yielding new macro-actions. By transferring or keeping the previously learned parameters fixed between action space augmentation, the agent can utilize the previously learned value characteristics. The weights of the final can then be relearned during the value learning stage. Again, we highlight the necessity of intra-macro learning in order to achieve sample efficient learning. Furthermore, it is necessary to maintain a separate experience buffer system in order to store transition tuples specific to inactive previously inferred macro actions. Finally, we highlight the importance of adapting the hyperparameters of the grammar inference algorithm. The length of the sampled trace is going to increase or decrease over the course of the learning procedure. Hence, regularization parameter of the  $k$ -Sequitur grammar inference algorithm has to be changed accordingly.



**Figure 2:** Online Grammar DQN with adaptive output.

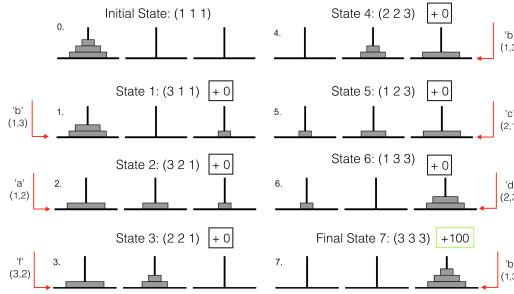
In order to accommodate the variable set of grammar-inferred skills the size of the DQN output layer has to be updated after every grammar learning step yielding new macro-actions. By transferring or keeping the previously learned parameters fixed between action space augmentation, the agent can utilize the previously learned value characteristics. The weights of the final can then be relearned during the value learning stage. Again, we highlight the necessity of intra-macro learning in order to achieve sample efficient learning. Furthermore, it is necessary to maintain a separate experience buffer system in order to store transition tuples specific to inactive previously inferred macro actions. Finally, we highlight the importance of adapting the hyperparameters of the grammar inference algorithm. The length of the sampled trace is going to increase or decrease over the course of the learning procedure. Hence, regularization parameter of the  $k$ -Sequitur grammar inference algorithm has to be changed accordingly.

## 5 Experiments

The goal of the experimental section of this work is to answer the following questions: (1) Does a grammar learned on optimal policy traces allow for rapid imitation learning? (2) Can CFG grammars be used in order to enhance curriculum as well as transfer learning? (3) Is online grammar inference and action space adaptation able to structure the exploration process of the HRL agent?

In order to illustrate our results we choose the  $N$ -disk Towers of Hanoi (ToH) environment (see left part of figure 3). The general game setting for  $N$  disks is as follows: In every episode of the game the agent is initialized in the tuple  $(1)_{i=1}^N$ . At each point in (discrete) time the agent transitions between states with the help of the following moves:  $\mathcal{A} = \{a : (1, 2); b : (1, 3); c : (2, 1); d : (2, 3); e : (3, 1); f : (3, 2)\}$ . The agent maximizes their expected cumulative discounted reward by reaching the final state  $(3)_{i=1}^N$  as quickly as possible. The problem is formulated as a sparse long-term credit assignment problematic. The size of the state space, on the other hand, grows exponentially,  $|S| = 3^N$  (all possible allowed orderings), and the optimal number of moves to solve this game is given by  $2^N - 1$ . A simple recursive procedure to solve the problem for all states in which the top  $N - n$  disks are already correctly ordered on the third pole is given by:

1. Move  $n - 1$  disks from source pole to auxiliary pole
2. Move the  $n$ -th disk from source pole to target pole
3. Move the  $n - 1$  disks that we left on auxiliary pole onto target pole



**Figure 3:** RL Formulation of the ToH Problem

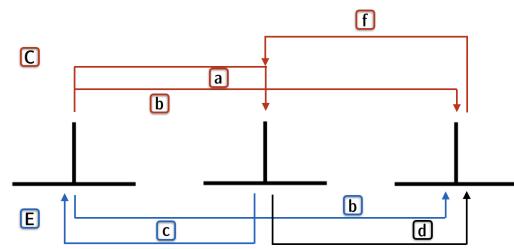
Identifying this underlying hierarchical principle and generalizing between different environments requires the agent to correctly identify their state in the underlying hierarchical action parse tree.

### 5.1 Learning with Expert Grammar and Transfer Grammar

Given action sequences of an expert the HRL agent can infer a set of grammar macros and augment their action space. Table 1 displays the grammar macros inferred from a trace of the optimal policy using the 2-Sequitur as well as the G-Lexis algorithm. The flattened production rule  $B \rightarrow CEd \rightarrow baabcd$  is visualized in figure 4 and captures the recursive nature learned by the grammar.  $C \rightarrow ba$  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. We can then augment the action space of the 5 disk agent in the following way:

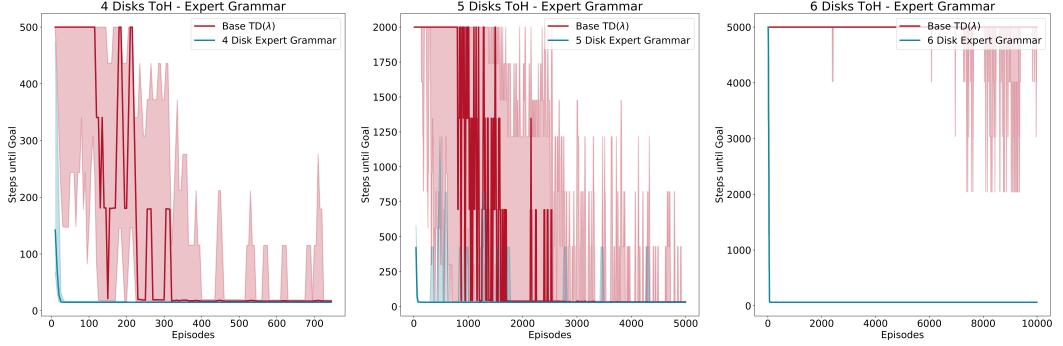
Trace ( $\vartheta$ ): baabcdbafecfbafbcdbcfecdbafbcdb				
	2-Sequitur		G-Lexis	
$\vartheta^{enc}$	BCDfBEfDdBb	BbafecfBbcfecdBb	$\vartheta^{enc}$	0.355
$ \vartheta $	0.355	0.516		
N	PR	$\mathcal{M}^{2-Seq}$	PR	$\mathcal{M}^{G-Lex}$
B	CEd	baabcd	-	baabcd
C	-	baf		
D	-	ec		
E	-	bc		

**Table 1:** ToH (5 disks) Grammar-Macro Construction



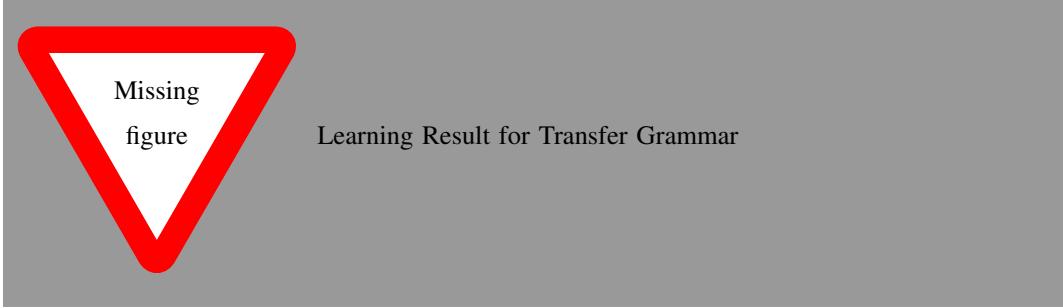
**Figure 4:** Visualization of the Grammar Macro "B".

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

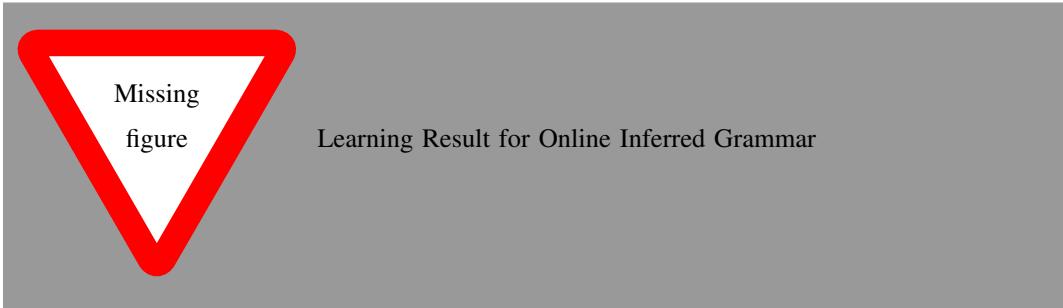


**Figure 5:** Learning Results for SMDP-Q-Learning with Expert Grammar Macros for 4, 5 and 6 Disk Environment. Median, 10th percentile and 90th percentile reported over 5 runs.

Figure 5 displays learning results for different SMDP-Q-Learning agents with macro-actions defined by the production rules inferred from a trace of the optimal policy. One can observe that the grammar macros greatly accelerate the learning progress of the agent. Furthermore, the variance of rollouts is greatly reduced due to increased robustness. Instead of using the optimal grammar macros for a specific environment, we are also interested in assessing how well the grammars learned for a more simplistic environment (i.e.  $N - 1$  disks).



## 5.2 Learning with Online Inferred Grammar



Rob: Add Grammar DQN experiments.

## 6 Discussion & Outlook

Motivated by a parallelism between the hierarchical generating processes of language and motion, we have derived multiple algorithmic approaches which exploit powerful grammatical inference frameworks to identify temporally-extended actions. At the center of this analysis was the formal notion of Semi-Markov Decision Processes and their capability to model stochastic waiting times between decisions. By sensibly defining temporally-extended actions and abstracting away unnecessary decision points, one is able to overcome the curse of dimensionality. In order to validate our proposed framework, we tested both approaches for an imitation learning as well as an online RL task in multiple environments. Our contributions can be summarized as follows: (1) The CFG approach to macro-actions extraction from flat production rules performs very well in both imitation and transfer learning. The agent can easily generalize from the inferred hierarchical structure and is able to increase the action learning speed drastically. (2) Alternating between grammar updates and learning action values is an effective way of both online learning of an optimal grammar as well as an optimal policy. The first grammar extraction and action space augmentation has the largest significant effect in the further learning procedure of the agent.

In future work we are interested in testing and extending our approach to physical and continuous (joint and velocity-bases state representations, e.g. MoJuCo) domains as well as model-based methods. Formal grammars are especially useful for languages with large terminal vocabulary. So far we have only experimented with discrete action spaces and single agents. Pastra and Aloimonos [27] note that social interactions of more than one agent can also be formulated within the notion of tool use. Hence, we are interested in possible applications to multi-agent RL and testing the scalability of our approach to real-life domains. Furthermore, our approach has only attempted to merge grammatical inference with one HRL algorithm. There remain many other promising frameworks such as the Hierarchies of Abstract Machines (HAMs, [26, 25]). HAMs define a hierarchy over finite state machines. This could naturally lead itself to automated identification via Hierarchical Hidden Markov Models [11]. Future work also has to further analyze the development of the inferred grammar throughout the learning process. Edit distances such as the Levenshtein and Jaro-Winkler distance provide two measures of string similarity which might be used to efficiently monitor the development of the inferred flat productions compared with the optimal grammar. Ultimately, we envision a form of dictionary of action which provides an expandable library of skills for Hierarchical Reinforcement Learning agents which act in diverse naturalistic environments. This could provide a major contribution to a key endeavor in general artificial intelligence: Life-long learning.

## Todo list

Figure: Learning Result for Transfer Grammar . . . . .	6
Figure: Learning Result for Online Inferred Grammar . . . . .	6
<span style="color: red;">█</span> Rob: Add Grammar DQN experiments. . . . .	6

## References

- [1] BACON, P.-L., J. HARB, AND D. PRECUP (2017): “The Option-Critic Architecture,” in *AAAI*, 1726–1734.
- [2] BAKKER, B. AND J. SCHMIDHUBER (2004): “Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization,” in *Proc. of the 8-th Conf. on Intelligent Autonomous Systems*, 438–445.
- [3] BRADTKE, S. J. AND M. O. DUFF (1995): “Reinforcement learning methods for continuous-time Markov decision problems,” in *Advances in neural information processing systems*, 393–400.
- [4] BRENNAN, J. R., E. P. STABLER, S. E. VAN WAGENEN, W.-M. LUH, AND J. T. HALE (2016): “Abstract linguistic structure correlates with temporal activity during naturalistic comprehension,” *Brain and language*, 157, 81–94.
- [5] CHARIKAR, M., E. LEHMAN, D. LIU, R. PANIGRAHY, M. PRABHAKARAN, A. SAHAI, AND A. SHELAT (2005): “The smallest grammar problem,” *IEEE Transactions on Information Theory*, 51, 2554–2576.
- [6] CHOMSKY, N. (1959): “A note on phrase structure grammars,” *Information and control*, 2, 393–395.
- [7] ——— (1959): “On certain formal properties of grammars,” *Information and control*, 2, 137–167.
- [8] DANIEL, C., H. VAN HOOF, J. PETERS, AND G. NEUMANN (2016): “Probabilistic inference for determining options in reinforcement learning,” *Machine Learning*, 104, 337–357.
- [9] DAYAN, P. AND G. E. HINTON (1993): “Feudal reinforcement learning,” in *Advances in neural information processing systems*, 271–278.
- [10] DING, N., L. MELLONI, X. TIAN, AND D. POEPPEL (2017): “Rule-based and word-level statistics-based processing of language: insights from neuroscience,” *Language, Cognition and Neuroscience*, 32, 570–575.
- [11] FINE, S., Y. SINGER, AND N. TISHBY (1998): “The hierarchical hidden Markov model: Analysis and applications,” *Machine learning*, 32, 41–62.
- [12] FLORENSA, C., Y. DUAN, AND P. ABBEEL (2017): “Stochastic neural networks for hierarchical reinforcement learning,” *arXiv preprint arXiv:1704.03012*.
- [13] FRANK, M. C., J. A. SLEMMER, G. F. MARCUS, AND S. P. JOHNSON (2009): “Information from multiple modalities helps 5-month-olds learn abstract rules,” *Developmental science*, 12, 504–509.
- [14] FRANK, S. L. AND M. H. CHRISTIANSEN (2018): “Hierarchical and sequential processing of language,” *Language, Cognition and Neuroscience*, 0, 1–6.
- [15] HENGST, B. (2002): “Discovering hierarchy in reinforcement learning with HEXQ,” in *ICML*, vol. 2, 243–250.
- [16] LEVELT, W. J. (2008): *An introduction to the theory of formal languages and automata*, John Benjamins Publishing.
- [17] MANNER, S., I. MENACHE, A. HOZE, AND U. KLEIN (2004): “Dynamic abstraction in reinforcement learning via clustering,” in *Proceedings of the twenty-first international conference on Machine learning*, ACM, 71.
- [18] MARCUS, G. F., K. J. FERNANDES, AND S. P. JOHNSON (2007): “Infant rule learning facilitated by speech,” *Psychological science*, 18, 387–391.
- [19] MARCUS, G. F., S. VIJAYAN, S. B. RAO, AND P. M. VISHTON (1999): “Rule learning by seven-month-old infants,” *Science*, 283, 77–80.
- [20] McGOVERN, A. AND A. G. BARTO (2001): “Automatic discovery of subgoals in reinforcement learning using diverse density,” in *ICML*, vol. 1, 361–368.
- [21] McGOVERN, A., R. S. SUTTON, AND A. H. FAGG (1997): “Roles of macro-actions in accelerating reinforcement learning,” in *Grace Hopper celebration of women in computing*, vol. 1317.
- [22] MENACHE, I., S. MANNER, AND N. SHIMKIN (2002): “Q-cut—dynamic discovery of sub-goals in reinforcement learning,” in *European Conference on Machine Learning*, Springer, 295–306.
- [23] NELSON, M. J., I. EL KAROUI, K. GIBER, X. YANG, L. COHEN, H. KOOPMAN, S. S. CASH, L. NACCACHE, J. T. HALE, C. PALLIER, ET AL. (2017): “Neurophysiological dynamics of phrase-structure building during sentence processing,” *Proceedings of the National Academy of Sciences*, 201701590.
- [24] NEVILL-MANNING, C. G. AND I. H. WITTEN (1997): “Identifying Hierarchical Structure in Sequences: A linear-time algorithm,” *CoRR*, cs.AI/9709102.
- [25] PARR, R. AND S. J. RUSSELL (1998): “Reinforcement learning with hierarchies of machines,” in *Advances in neural information processing systems*, 1043–1049.
- [26] PARR, R. E. (1998): *Hierarchical control and learning for Markov decision processes*, University of California, Berkeley Berkeley, CA.
- [27] PASTRA, K. AND Y. ALOIMONOS (2012): “The minimalist grammar of action,” *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 367, 103–117.
- [28] SIMSEK, O., A. P. WOLFE, AND A. G. BARTO (2004): “Local graph partitioning as a basis for generating temporally-extended actions in reinforcement learning,” in *AAAI Workshop Proceedings*.
- [29] SIYARI, P., B. DILKINA, AND C. DOVROLIS (2016): “Lexis: An optimization framework for discovering the hierarchical structure of sequential data,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 1185–1194.
- [30] SIYARI, P. AND M. GALLÉ (2016): “The Generalized Smallest Grammar Problem,” in *International Conference on Grammatical Inference*, 79–92.
- [31] SMITH, M., H. VAN HOOF, AND J. PINEAU (2018): “An Inference-Based Policy Gradient Method for Learning Options,” in *Proceedings of the 35th International Conference on Machine Learning*, 4710–4719.
- [32] STOLLE, M. AND D. PRECUP (2002): “Learning options in reinforcement learning,” in *International Symposium on abstraction, reformulation, and approximation*, Springer, 212–223.
- [33] STOUT, D., T. CHAMINADE, A. THOMIK, J. APEL, AND A. A. FAISAL (2018): “Grammars of action in human behavior and evolution,” *bioRxiv*, 281543.
- [34] SUTTON, R. S., D. PRECUP, AND S. SINGH (1999): “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning,” *Artificial intelligence*, 112, 181–211.
- [35] VEZHNEVETS, A. S., S. OSINDERO, T. SCHAUL, N. HEESS, M. JADERBERG, D. SILVER, AND K. KAVUKCUOGLU (2017): “Feudal networks for hierarchical reinforcement learning,” *arXiv preprint arXiv:1703.01161*.

## **Supplementary Material**

### **Agent Architecture and Hyperparameters**

### **Notes on Reproduction**

Please clone the repository <https://github.com/RobertTLange/action-grammars-hrl> and follow the instructions outlined below: