Using Neural Networks to Control Glut in the Active Logic Machine

Justin D. Brody [1] Bobby Austin [1] Omar Khater [1] Christopher Maxey [2] Matthew D. Goldberg [2] Timothy Clausner [2] Darsana Josyula [3] Donald Perlis [2]

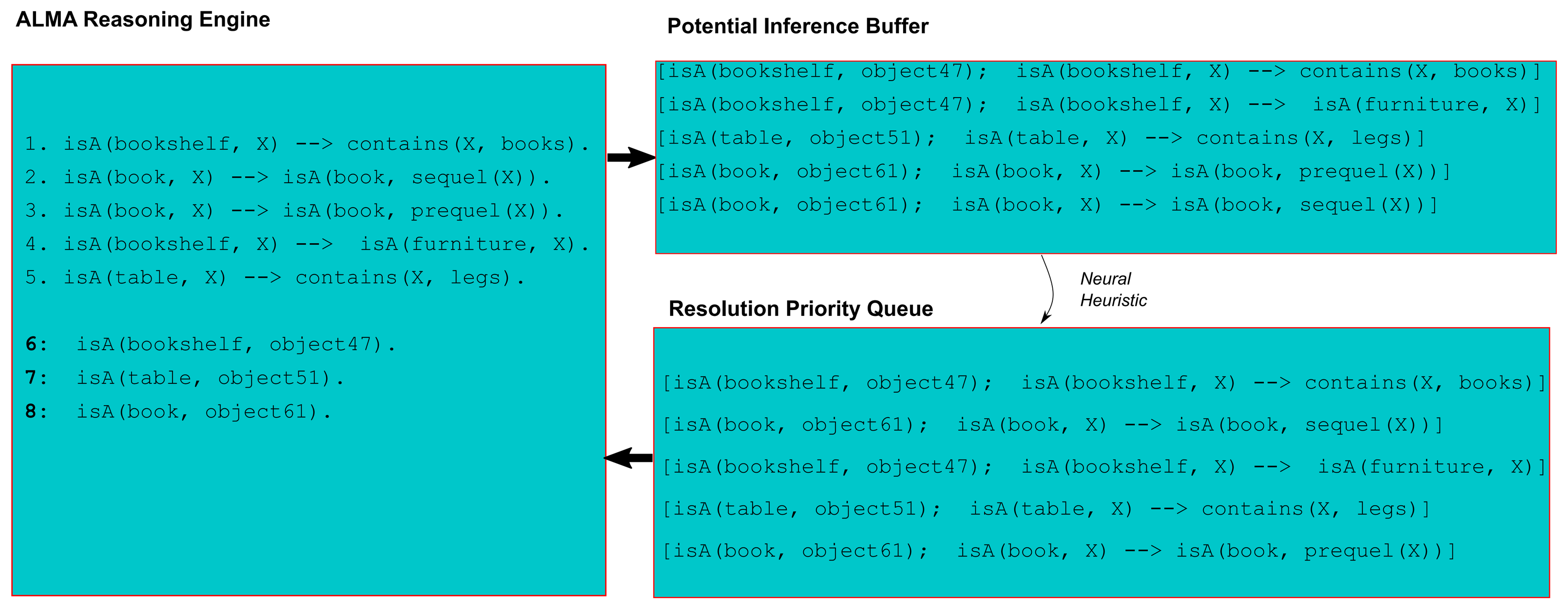
[1]Franklin and Marhsall College [2]University of Maryland Institute for Advanced Computer Studies [3]Bowie State University

Introduction

Inferential glut refers to the problem that symbolic reasoners tend to produce vastly more deductive information than they can tractably process We examine two-types We design a neuro-symbolic architecture that reduces both kinds of glut in the Active Logic Machine reasoning with resolution.

The Architecture

Idea: As potential inferences are recognized by the system, they are assigned a priority by a neural network. Potential inferences are then processed by their priority.



Pre-Inferential Glut Control

Pre-inferential glut comes from mis-labeling non-unifying pairs of sentences as unifying.

We worked with an initial knowledge base having a single sentence of the form: $f(x_0, y_0, t) \wedge f(b, y_1, t) \rightarrow g(x_0, b, y_0, t)$ along with 50 observations of the form $f(a, d_i, i), f(b, d_i, i)$ for i an integer with $0 \leq i < 50$ and d_i random integers with $0 \leq d_i < 100$. Logically, there are 100 conclusions that can be derived from these initial sentences .

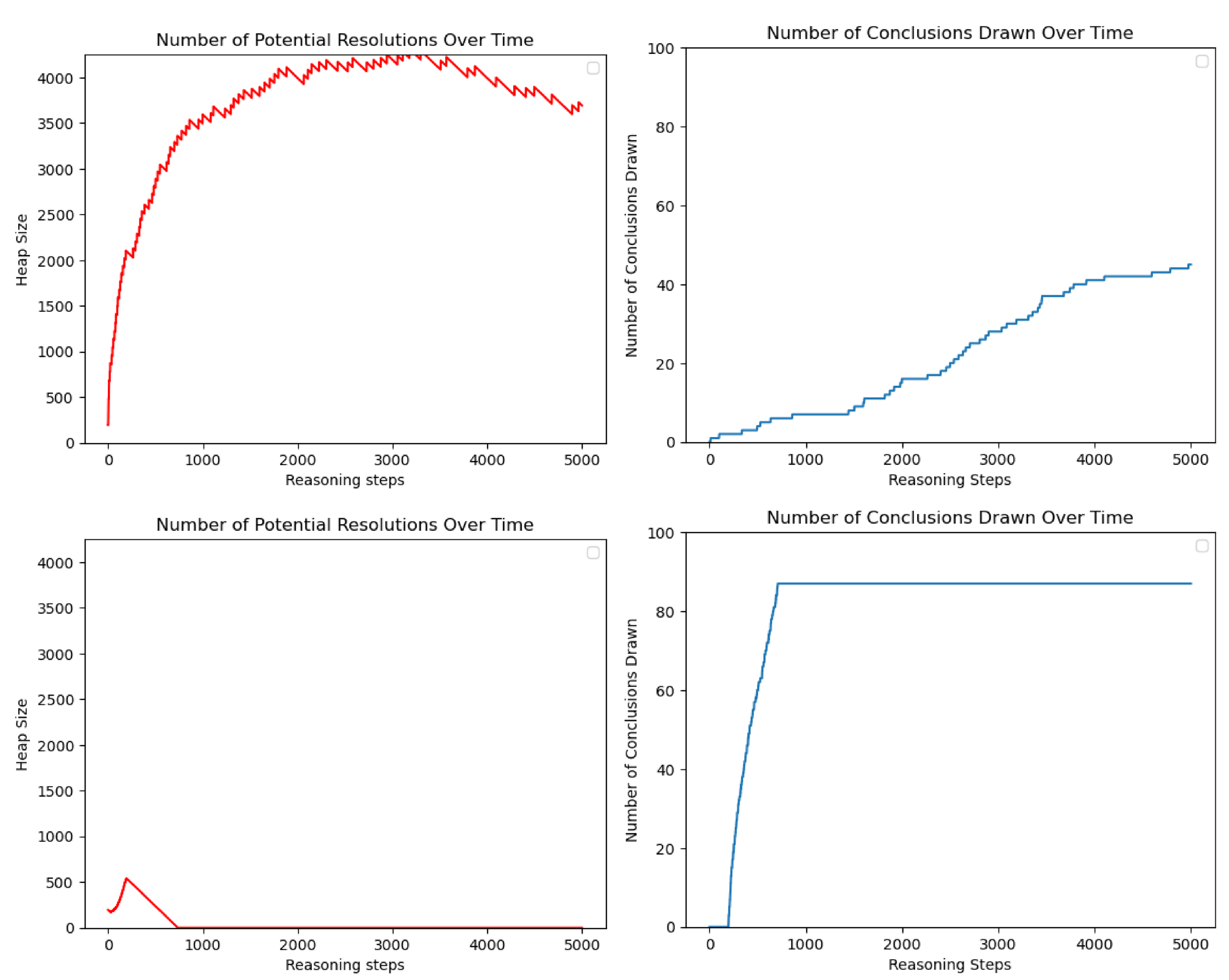


Figure 1. After training, Heuristic ALMA is able to keep the size of the priority queue small and draw most of the inferences available to it.

Inferential Glut Control

Inferential glut comes from pairs of sentences combining via resolution but not leading to valuable new information. To control it, we set up the problem of glut control as a reinforcement learning problem and use deep Q-learning.

We represented every logical sentence as a feature-graph corresponding to a syntax tree.

nID	a	book	left	right	if	and	or	func	var
0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1
3	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	1

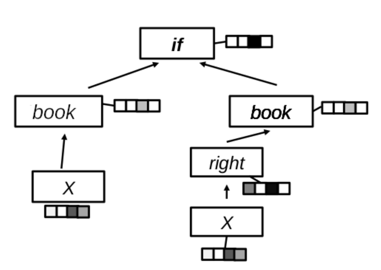


Figure 2. Vectorization

We encode potential inferences as pairs of such graphs which become inputs to a deep network We trained our system using a network structure identical to that used in the pre-inferential scenario above. Our reward function assigned a reward equal to the total number of new instances of the function right in the knowledge base

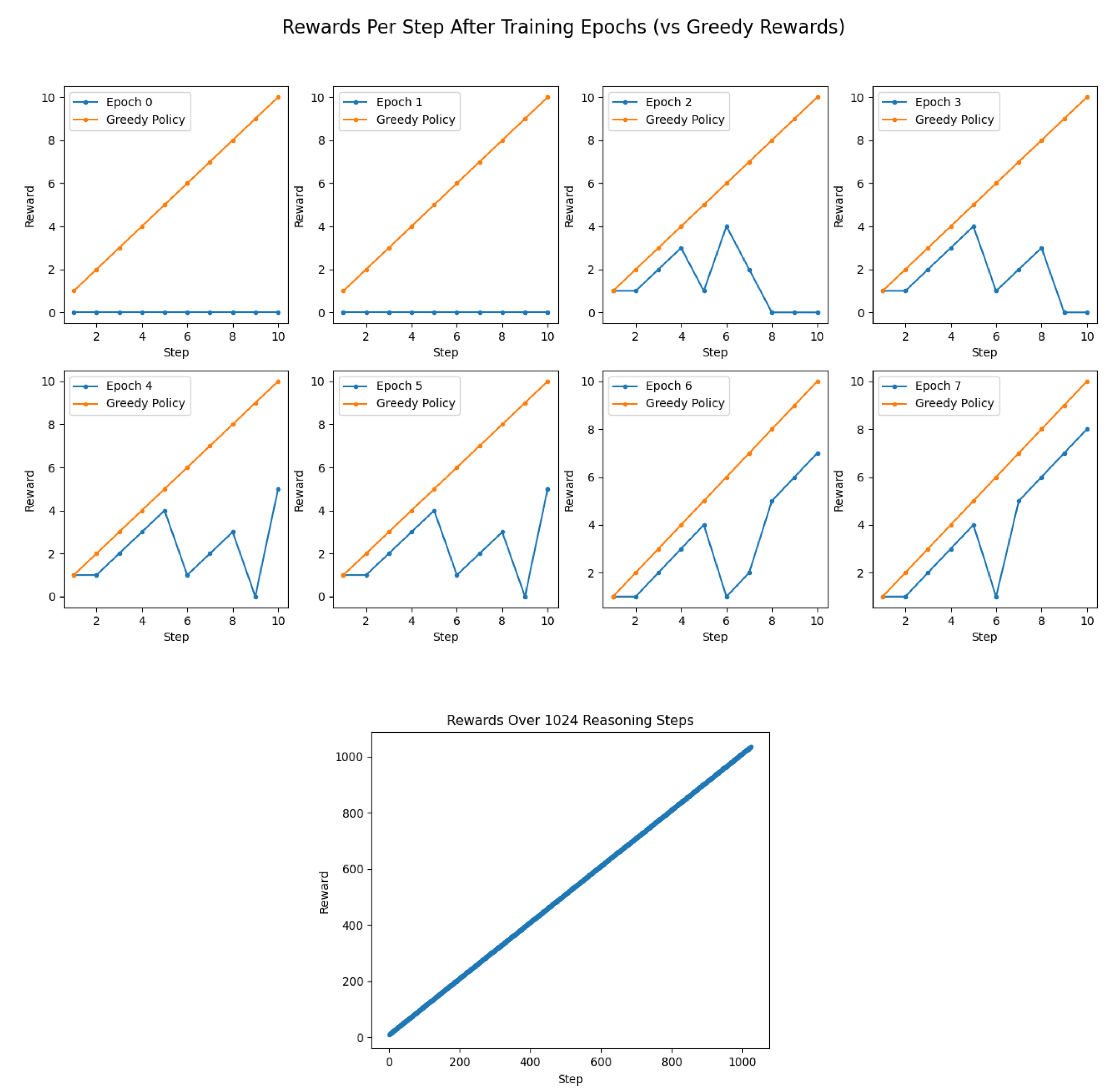


Figure 3. Training Results: The strategy learned for 10 steps extrapolates to a large number of steps.

References

[Elgot-Drapkin and Perlis(1990)] J. J. Elgot-Drapkin, D. Perlis, Reasoning situated in time i: Basic concepts, Journal of Experimental & Theoretical Artificial Intelligence 2 (1990) 75–98.

[Elgot-Drapkin et al.(1999)Elgot-Drapkin, Kraus, Miller, Nirkhe, and Perlis] J. Elgot-Drapkin, S. Kraus, M. Miller, M. Nirkhe, D. Perlis, Active logics: A unified formal approach to episodic reasoning, Technical Report, 1999.

[Anderson et al.(2008)Anderson, Gomaa, Grant, and Perlis] M. L. Anderson, W. Gomaa, J. Grant, D. Perlis, Active logic semantics for a single agent in a static world, Artificial Intelligence 172 (2008) 1045–1063.

[Purang(2001)] K. Purang, Alma/carne: implementation of a time-situated meta-reasoner, in: Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001, IEEE, 2001, pp. 103–110.

[Russell and Norvig(2020)] S. Russell, P. Norvig, Artificial intelligence: a modern approach, Prentice Hall, 2020.

[Parikh(1987)] R. Parikh, Knowledge and the problem of logical omniscience., in: ISMIS, volume 87, Citeseer, 1987, pp. 432–439.

[Wang et al.(2017)Wang, Tang, Wang, and Deng] M. Wang, Y. Tang, J. Wang, J. Deng, Premise selection for theorem proving by deep graph embedding, arXiv preprint arXiv:1709.09994 (2017).

[Huang et al.(2018)Huang, Dhariwal, Song, and Sutskever] D. Huang, P. Dhariwal, D. Song, I. Sutskever, Gamepad: A learning environment for theorem proving, in: International Conference on Learning Representations, 2018.

[Polu and Sutskever(2020)] S. Polu, I. Sutskever, Generative language modeling for automated theorem proving, arXiv preprint arXiv:2009.03393 (2020).

[Rawson and Reger(2019)] M. Rawson, G. Reger, A neurally-guided, parallel theorem prover, in: International Symposium on Frontiers of Combining Systems, Springer, 2019, pp. 40–56.

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

[Mnih et al.(2015)Mnih, Kavukcuoglu, Silver, Rusu, Veness, Bellemare, Graves, Riedmiller, Fidjeland, Ostrovski et al.] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, nature 518 (2015) 529–533.

[Kipf and Welling(2016)] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[Wang et al.(2019)Wang, Yu, Zheng, Gan, Gai, Ye, Li, Zhou, Huang, Ma et al.] M. Wang, L. Yu, D. Zheng, Q. Gan, Y. Gai, Z. Ye, M. Li, J. Zhou, Q. Huang, C. Ma, et al., Deep graph library: Towards efficient and scalable deep learning on graphs. (2019).