Log based behavioral system model inference using reinforcement learning

TU Delft – Research Project CSE300

Author: Pandelis Symeonidis
p.l.symeonidis@student.tudelft.nl

Supervisors: Mitchell Olsthoorn, Annibale Panichella
{m.j.g.olsthoorn, a.panichella }@tudelft.nl

1. PROBLEM

Model behavior of large, real-time systems

Useful for [1]:

- System comprehension
- Test case generation
- Documentation

Existing techniques:

- Profiling – affects real-time behavior
- Source code inference- unscalable
- Log –inference – current techniques do not scale

2. PROPOSED SOLUTION

Log-based, FSM inference using reinforcement learning (RL)

RQ: "How effective is reinforcement learning for inferring a concise and accurate state model of code behavior?"

- Accuracy of the model?
  - F1 score (recall & precision)
  - Specificity
- Conciseness of model?
  - Model compression
- Scalability of approach?

3. CONTEXT

- XRP Ledger
- Decentralized, distributed, real-time system
- Focus on the **consensus algorithm**
  - Filter logs
  - Split consensus rounds (traces)

XRP logo

4. APPROACH

1) Parse logs to initial model -> 2) Condense to final model by merging states

1. LOG ANALYSIS

- Syntax Tree: log template identification
- Unique State Graph (USG): parse log traces to initial model

How merge states?

2. FLAPPY BIRD APPROACH

Apply Q-Learning [2] by traversing model

Goal: Learn to merge states based on the topology of the surrounding states.

States:

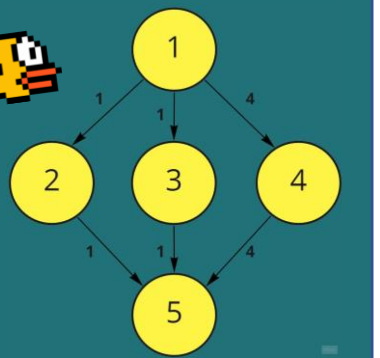
- # outgoing edges
- Outgoing edges entropy

Actions:

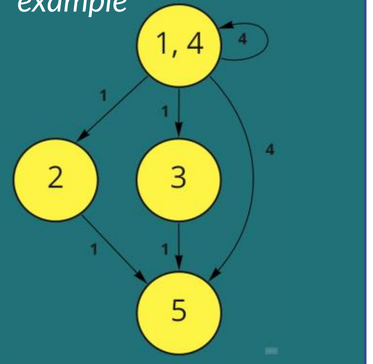
- Don't merge
- Merge most frequent
- Merge second most frequent, etc.

Reward agent for merging nodes trading off **accuracy** for **compression**

End episode when reward is negative



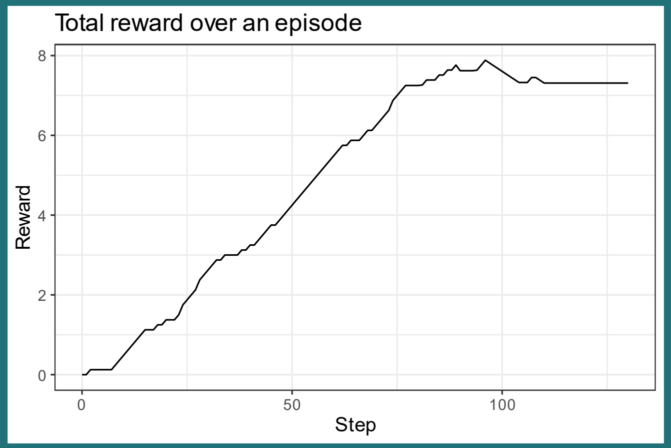
1. Unique state graph example



2. USG merging state 1 and 4

5. RESULTS

- Inside episode total reward increases providing promising solutions (models)
- Across episodes no significant sign of learning



- Approach scales linearly with input logs

6. DISCUSSION

- Assumptions
  - Each log template is unique in the codebase
  - The surrounding region of each state is enough to decide how to merge
- Assumptions hold on a case-by-case basis
- State space not large enough
- Q-values cannot generalize to whole model

7. CONCLUSION & FUTURE WORK

- Promising results in each episode
- Q-values might not generalize in this case

- Fine tune hyper-parameters
- Extend/experiment with state space + action space
- Fine tune reward function

Github repo

1. Damas, C., Lambeau, B., Dupont, P., & Van Lamsweerde, A. (2005). Generating annotated behavior models from end-user scenarios. *IEEE Transactions on Software Engineering*, *31*(12), 1056-1073.
2. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.