Graph Contrastive Pre-training for Effective Theorem Reasoning

Zhaoyu Li, Binghong Chen, Xujie Si

1. Background & Motivation

Automated reasoning over mathematics proofs is an intriguing challenge:
- it requires machines to understand sophisticated **high-order logic** for **reasoning**.

Interactive theorem proving

Interactive theorem proving (ITP) allows humans to develop **formal proofs** of mathematical theorems by interacting with a computer system (e.g. Coq[1]).

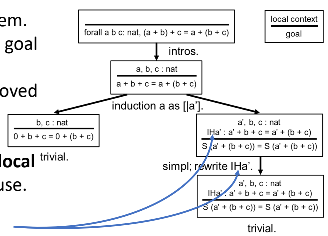
Define **mathematical objects**. →
```
Inductive nat :=
  | O : nat
  | S : nat -> nat.
```

State a **theorem** to prove. →
```
Theorem add_assoc:
  forall a b c : nat,  (a + b) + c = a + (b + c)
```

Prove the given theorem by entering a sequence of commands called **tactics**. →
```
Proof.
  intros.
  induction a as [|a'].
  + trivial.
  + simpl; rewrite IHa'. trivial.
Qed.
```

Proving procedure in Coq

- The initial **goal** is the given theorem.
- Tactics decompose the current goal into several (can be 0) sub-goals.
- The theorem is completely proved when there is no sub-goals left.
- Each goal shares a same **environment** and has an unique **local context** with useful **premises** to use.
- Premises can be used as **tactic arguments** to simplify the proof.



Why we use machine learning

Drawback of Coq ITP:
- **Labor-intensive**
- **Non-trivial expertise**

CompCert[2] compiler certification project:
- **Six PhD-years**
- More than **100,000 lines** of proof script

Iris[3] concurrent program verification Project:
- **Five PhD-years**
- More than **143 Coq files**

+ enough data

Tactic prediction is to automate this proof procedure.

...

2. Problem & Challenges

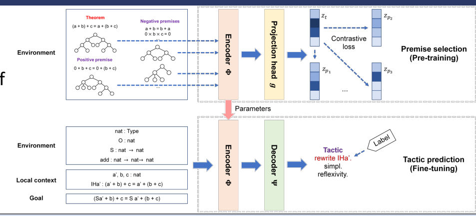
Input:
- A goal g to be proved and some existing premises $p_1, p_2, \cdots, p_N$ in the environment or local context that can be used.

Output:
- A tactic with its arguments (if it can carry).

Challenges:
- How to leverage **human expert insight** to design our model?
- How to represent theorems and premises effectively?
- How to predict the tactic and its arguments?

3. Our Approach

The overview of **NeuroTactic**'s framework



Premise selection pre-training task

Existing methods:
- Focus on **directly predicting a sequence of appropriate tactics** that presumably prove the given theorem.

Human experts:
- Often speculates a high-level plan (e.g. **figure out lemmas or premises that are going to be used)** before writing down any tactics.

To learn better representations of theorems and premises

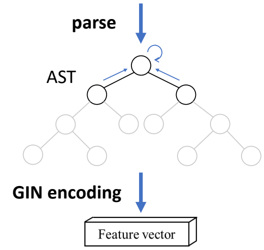
Proposed **premise selection** pretext task:
- **Select relevant statements** that are useful for proving a given theorem.

Theorem representations and encoding

Surface-level representation: →
```
Theorem add_assoc:
  forall a b c : nat,  (a + b) + c = a + (b + c)
```
- Dependent types
- Various syntax sugars
- Very flexible grammar

Kernel-level representation: →
- Simple grammar
- Uniform representation
- Represented by **ASTs**

GIN[4] embedding:
- Tree as undirected graph
- Syntax roles as node features
- More powerful than **TreeLSTM**[5]

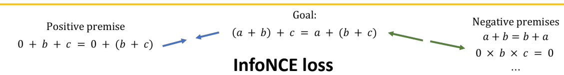


Semantic-guided graph contrastive pre-training

Observation:
- The relevance between theorems and premises depends on their **semantic relevance**.
- Practical premises that can be used as **tactic arguments** often **share the same semantic components with the current goal.**

a', b, c : nat
IHa' : a' + b + c = a' + (b + c)
S (a' + (b + c)) = S (a' + (b + c))

We employ the existing theorems and premises as our learning pairs:
- Positive pairs: The goal and premises that can be used as tactic arguments in the environment or local context
- Negative pairs: The goal and other premises



Tactic prediction decoder

We use the same decoder in ASTactic[6]:
- Conditioned on embeddings from our encoder, the decoder generates tactic and its arguments by selecting production rules and argument tokens following a context-free grammar (CFG) for its tactic space.

4. Experimental Evaluation

Experimental setup

Premise selection:
- Proposed **PremiseGym** dataset: **10533** instances for training, **3783** instance for testing.
- Each instances has a goal to decompose, a positive premises and more than 8 negative premises.

Tactic prediction:
- **CoqGym**[6] dataset: **189824** tactics for training, **78494** tactics for testing.

Experimental results

Tactic prediction results

Project	ASTactic	NeuroTactic		Total
	TreeLSTM	TreeLSTM+GCL	GIN+GCL	
PolTac	**79**	59	59	190
UnifySL	677	713	**722**	2865
angles	**10**	7	6	199
buchberger	**34**	33	**34**	299
chinese	97	**127**	126	462
coq-library-undecidability	196	**233**	224	3181
coq-procrastination	0	0	0	1
coqoban	0	0	0	7
coqrel	20	19	**23**	94
coquelicot	281	281	**283**	3498
dblib	73	86	**87**	371
demos	17	15	**20**	192
dep-map	**16**	9	10	142
disel	3	**4**	**4**	47
fermat4	5	10	**11**	45
fundamental-arithmetics	**127**	122	**127**	420
goedel	**999**	993	995	6640
hoare-tut	3	3	3	27
huffman	5	3	**6**	108
jordan-curve-theorem	5470	7352	**7531**	28672
tree-automata	3778	3761	**3809**	15201
verdi	**239**	232	235	1917
verdi-raft	1714	1747	**1802**	11063
weak-up-to	0	**2**	1	52
zchinese	40	47	**52**	247
zfc	67	**83**	72	461
zorns-lemma	337	337	**360**	2093
Total	14287	16278	**16602**	78494

Premise selection results

TreeLSTM correctly selects 1399 premises (**36.98%**) for the given theorems. Our encoder successfully predicts **1704** premises (**45.04%**), which obtains more than **21.8%** relative improvement.

1. Barras, B. et al. "The Coq proof assistant reference manual: Version 6.1." PhD thesis, Inria, 1997.
2. Leroy, X. et al. "Compcert - a formally verified optimizing compiler." In ERTS 2016: Embedded Real Time Software and Systems, 8th European Congress, 2016.
3. Jung, R. et al. " Iris from the ground up: A modular foundation for higher-order concurrent separation logic." Journal of Functional Programming, 28, 2018.
4. Xu, K. et al. "How powerful are graph neural networks?" arXiv preprintarXiv:1810.00826, 2018.
5. Tai, K. S. et al. "Improved semantic representations from tree-structured long short-term memory networks." arXiv preprint arXiv:1503.00075, 2015.
6. Yang, K. et al. "Learning to prove theorems via inter-acting with proof assistants." In International Conference on Machine Learning, pp. 6984–6994. PMLR, 2019.

McGill University & Mila - Quebec AI Institute & Georgia Institute of Technology

Contact: zhaoyu.li@mail.mcgill.ca; binghong@gatech.edu; xsi@cs.mcgill.ca