Theorem proving and 2-player games

Fan Zheng

May 2, 2023

Chess lines

- 1. e4
  - 1... e5
  - 1... e6
  - 1... c5
  - ...
- 1. d4
  - 1... d5
  - 1... Nf6
  - ...
- ...

Red nodes: Pick the most favorable move

Blue nodes: Consider all responses, especially strong ones

Theorem proving

- Library lemma 1
  - Hypothesis 1
  - Hypothesis 2
  - Hypothesis 3
- Library lemma 2
  - Hypothesis 1
  - Hypothesis 2
- ...

Red nodes: Pick the easiest approach

Blue nodes: Verify all hypotheses, especially hard ones

Two approaches

Fashionable approach
— MCTS $+$ Neural network evaluation $=$ Alphazero solves chess
GOFAI approach
— MCTS $+ X = Y$ solves math?
(I consider MCTS GOFAI because it appeared before deep NNs. Also it's completely deterministic, contrary to what its name suggests.)

Playground: Metamath

Plain syntax:

$$(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$$

Simple proof steps:

Step	Hyp	Ref	Expression
1		ax-1	$.\vdash (\phi \rightarrow (\phi \rightarrow \phi))$
2		ax-1	$.\vdash (\phi \rightarrow ((\phi \rightarrow \phi) \rightarrow \phi))$
3	1,2	mpd	$\vdash (\phi \rightarrow \phi)$

Easy verification: $<$ 1k lines of code

Playground: Propositional logic

Complete: All true sentences provable.
Decidable: $\exists$ algorithm telling if a sentence is true ($\leftrightarrow$ provable).
Hard: 3SAT is NP complete.
Tools: SAT solvers.

Solver vs verifier

	Solver	Verifier
Function	Check validity, finds proof	Verifies proof
Deduction system	Domain-customized	Same throughout
Kernel	Large and complex	Small and simple
Soundedness	May be buggy	Easily verified

Get the best of both worlds

Trace the execution of the solver and translate the proof step by step — tedious, because, among other things, you need m solvers $\times n$ verifiers $= mn$ translators

A better way

MCTS + $X = Y$ solves math?

X = solver as evaluator!

- X evaluates each subgoal and discard unprovable ones.
- MCTS selects the most natural/idiomatic/human proof.

For m solves and n verifiers, only needs $m + n$ adaptors.

Test results

Out of the 1846 propositional theorems in Chapter 1 of set.mm,

Search at 2^{10} nodes	# proven	%
Untampered	**1686**	**91.3**
Bad MCTS parameter	1289	69.8
No SAT	480	26.0

Optimizations

- Loop detection: $A \leftarrow B \leftarrow A$ is counted as a loss.
- Hypothesis simplification (using SAT):
  - Reduces # provable subgoals to choose from.
  - Still retains the ones actually needed for the proof.
- Evaluation(subgoal) =

$$\begin{cases} 1 \text{ (win) }, & \text{if subgoal} \in \text{hypotheses or is proven}, \\ -1 \text{ (loss) }, & \text{if SAT(!subgoal)} = \text{satisfiable}, \\ \frac{1}{|\text{subgoal}| + |\text{hypotheses}| + 1}, & \text{otherwise.} \end{cases}$$

- Parameters:
  Exploration for our moves = 0.001,
  Exploration for their moves = 0.

Future directions

- Generalizations: Propositional calculus (done) $\rightarrow$ Predicate calculus $\rightarrow$ ZFC set theory $\rightarrow \mathbb{R}/\mathbb{C} \rightarrow$ Analysis/algebra/topology
- Abstractions: Boolean, first order, ...
- Planning: Formalizing conjectures as intermediate subgoals
- Evaluating sentences in undecidable theories:

$$\exists\forall\exists\forall\ldots$$

Red nodes: Our move, pick favorable instances
Blue nodes: Their move, check many instances