Neural Theorem Proving on Inequality Problems

Yuhuai Wu*, Albert Jiang*, Roger Grosse, Jimmy Ba

1 Introduction

We present an attempt to prove rudimentary inequality theorems as solving sequential decision making problems. Our contributions include 1., designing an inequality theorem generator that can sample non-trivial inequality theorems of arbitrary length, given a set of axioms 2. demonstrating various degree of generalizations of a graph neural network based learning agent. We believe the proposed dataset can be used as a testbed for machine learning methods on theorem proving.

2 Proving Inequalities as a Markov Decision Process

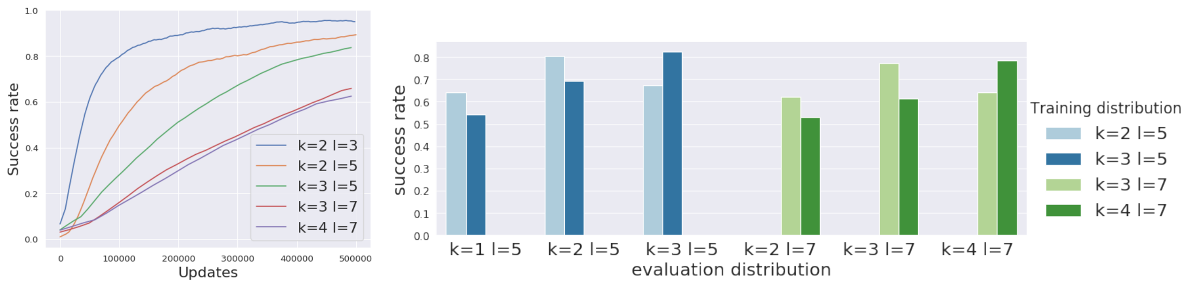
Similar to Huang et al. [2018], Bansal et al. [2019], Yang and Deng [2019], we model proving inequality theorems as a Markov Decision Process. Different from the approach by Fawzi et al. [2019] which models polynomial inequalities as optimization problems, our framework is more general and can be easily extended to other kinds of mathematical theorem proving tasks. We chose inequality problems because of their simplicity and analytic transparency, while embodying the general difficulty of theorem proving. The state space consists of a set of known facts (premises and proved sentences), and a set of goals (the formulae to prove). The action space is a tuple of an axiom and its arguments. The axiom set has all the axioms to define an ordered ring, plus a few composed axioms to reduce redundancy. All axioms are listed in Appendix A.

Our proof system allows the agent to prove in both forward and backward directions. Given the axiom and entities chosen, the proof system first constructs a formula p representing the premise, and a formula c representing the conclusion. If the premise p exists in the current set of ground truth, then the conclusion c is proved and c will be added to the set of known facts in the next state; this is proving in the forward direction. On the other hand, if the goal set contains the conclusion, this suggests we can convert the goal from c to p. We hence delete the c from goal set and add p to it; this is proving in the backward direction. When the goal set is a subset of the set of facts, the proof is finished and the agent succeeds.

3 Dataset

One way to generate theorems is to randomly sample a sequence of actions and apply them to the initial state s_0, transforming it to a new state s. Each of the proved facts in s with the minimal premises needed to prove it can be treated as a new theorem. However, to generate a new conclusion, the premise needs to be satisfied a priori. It is therefore not easy to sample a long and interesting theorem from randomly applied actions. The set of theorems generated this way are skewed towards using axioms whose premises are easy to satisfy, and usually require not more than 2 steps to prove.

Inequality theorem generator Hence, one of the main contribution of this paper is to design a theorem generator that is able to sample non-trivial and long theorems. We overcome those previously mentioned challenges by writing a production rule for each axiom. This not only ensures that new conclusions can be generated by constructing premises on the fly, but also to chain up previous proof steps to form longer proofs. Our strategy is to start with a simple formula (e.g. $a = a$) and iteratively transform it to a more complicated one to prove. The synthesis algorithm takes in a list of axioms, and two integer parameters k and l, and generates a theorem distribution. Each theorem in the distribution can be proved using k unique



(a) (b)

Figure 1: (a) shows learning curves evaluated on the same theorem distribution. (b) shows generalization performances across various k while keeping l fixed.

axioms from the list of axioms, and within total proof steps l. We provide the pseudo-code for the synthesis algorithm in Algorithm 1. We also constrain the length of the sampled expressions to be short, so as to make the theorems more natural.

Six Dimensions of Generalization Our motivation of creating the dataset is to use it as a tool to answer research questions for learning methods on theorem proving. The most essential challenge in theorem proving learning is to be able to prove new theorems that has not been seen during training, i.e., the problem of generalization. Specifically, given a fixed set of axioms, we consider testing agent's generalization ability on unseen theorems whose variations are resulted from 1. the randomness from theorem distribution, 2. varying the complexity of initial conditions 3.varying the orders in which axioms are applied, 4. varying the combinations of axioms used, 5. varying the number of unique axioms k, and 6. varying the length of the proof l. Notice that the first generalization is still within *i.i.d.* regime, namely, the unseen theorems are sampled from the same theorem distribution as in training. In contrast, the second and the third generalization considers non *i.i.d.* generalization, an essential characterization of the challenge of in learning theorem proving.

4 Experiments

We now present our results on generalization along some generalization dimensions. We converted each formula to its computational graph, and used a graph neural network based agent to encode the state, and an autoregressive model to propose actions. We used the proposed inequality theorem generator to generate theorems as well as their proofs as training data for imitation learning. We generated data in an online fashion to reduce overfitting. All of these agents were trained for 500000 updates. For offline evaluation, we calculated the success rate by running the agents for 10 steps on 1000 theorems sampled from a particular distribution.

The generalization results over the first dimension when fixing k and l is demonstrated in Fig 1 (a). We observe that our learning algorithm was able to improve performances over all pairs of k and l, indicating generalization over the same distribution is successful.

To test the agent's generalization ability over k, we evaluated our agents on theorem distribution with varying number of axioms k, while the length of the proof l was kept unchanged. The results are presented in Fig 1 (b). We observe that the agent that was trained on theorem distribution $k = 2$, $l = 5$ was able to solve to unseen theorems from distribution $k = 1, 3$ and $l = 5$, with a $13 - 16\%$ performance degradation. Similarly for the other three agents, all suffered from $10 - 30\%$ performance degradation. The largest drop in performance happened in evaluating the agent trained on $k = 3$, $l = 5$ and evaluated on $k = 1$, $l = 5$. It is surprising because $k = 1$, $l = 5$ may seem to be an easier theorem distributions than $k = 3, l = 5$, but the agent did worse on 29% worse on easier theorems, indicating poor non-*i.i.d.* generalization.

Lastly, to test agent's ability to prove longer proofs, we evaluated our agents on all theorem distribution shown in the Table.(1). We first observe that all agents suffered performance degradation when the length of the proof is increased. Agents trained on simple theorem distribution such as $k = 2$ $l = 3$ could not

Table 1: Generalization performances of agents trained on various theorem distributions.

Training on	Evaluation								
	k=1 l=1	k=1 l=3	k=2 l=3	k=2 l=5	k=3 l=5	k=3 l=7	k=3 l=9	k=4 l=7	k=4 l=9
k=2 l=3	92.9%	84.7%	**98.0%**	0	4.1%	0	0	0	0
k=2 l=5	94.9%	84.7%	93.9%	73.5%	68.4%	28.6%	5.1%	22.4%	4.1%
k=3 l=5	86.7%	86.7%	92.9%	66.3%	83.7%	38.8%	4.1%	43.9%	6.1%
k=3 l=7	95.9%	80.6%	90.0%	84.7%	84.7%	**70.4%**	**58.2%**	72.4%	**55.1%**
k=4 l=7	96.9%	85.7%	90.8%	64.3%	82.7%	56.1%	40.8%	**75.5%**	46.9%
k=4 l=9	91.8%	80.6%	78.6%	59.2%	62.2%	50.0%	41.8%	68.4%	42.9%
curriculum	**98.0%**	**95.9%**	94.9%	**92.9%**	**86.7%**	66.3%	21.4%	63.3%	28.6%

generalize to any problems beyond length larger than 3. In contrast, agents trained on more difficult theorem distributions, such $k = 4$, $l = 9$, can generalize to easier theorem distributions with good performances. Most remarkably, the agent trained on $k = 3$ $l = 7$ was able to achieve quite impressive generalization on longer problems $k = 3$ $l = 9$ and $k = 4$ $l = 9$, even surpassing the performances of those agents that are trained on these training distributions. This indicates a curriculum agent that learns easier theorem distribution could help prove harder theorems. Therefore, in additional to agents trained on single theorem distribution, we also trained an agent on all theorem distributions consisting of $k = 1$ $l = 3$, $k = 1$ $l = 5$, $k = 2$ $l = 3$, $k = 2$ $l = 5$, $k = 3$ $l = 5$, named **curriculum** in the results. We can see that it is indeed able to beat most of the agents in the distribution it was trained on, and almost matched the $k = 3$ $l = 7$ agent on $k = 3$ $l = 7$, demonstrating the benefits brought by curriculum learning.

References

Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. HOList: An environment for machine learning of higher order logic theorem proving. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 454–463, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/bansal19a.html.

Alhussein Fawzi, Mateusz Malinowski, Hamza Fawzi, and Omar Fawzi. Learning dynamic polynomial proofs. In *Advances in Neural Information Processing Systems*, pages 4181–4190, 2019.

Daniel Huang, Prafulla Dhariwal, Dawn Song, and Ilya Sutskever. GamePad: A Learning Environment for Theorem Proving. *arXiv preprint arXiv:1806.00608*, 2018.

Kaiyu Yang and Jia Deng. Learning to prove theorems via interacting with proof assistants. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6984–6994, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/yang19a.html.

A Inequality Theorem Generator

For each step of the synthesis program, we will sample an axiom from the given list, and we initiate three procedures, `NEW`, `EXT` and `SUB` associated to the sampled axiom. The `NEW` procedure produce any new premise so that the given axiom can be applied in the synthesized proof. The `EXT` procedure then proceeds to produce the entities as input to the axiom. Lastly, in order to chain up the previous proving steps and the new step, one may be required to do an extra substitution step called `SUB` procedure. The resulting new formula then is used to replace l.

Algorithm 1 Inequality Theorem Generator

Input: Set of all available axioms A, cardinality of axioms to use k, desired length of proof l
Output: Synthesized goal g, a set of premises H
  Set of theorems to use $A_k \leftarrow \text{sample}(A, k)$
  Initialize set of used theorems $A' \leftarrow \emptyset$, premise set $H \leftarrow \emptyset$
  Initialize fact $f \sim \text{Uniform}(\{a = a, b = b, c = c\})$
  Initialize length counter $c \leftarrow 0$
  while $c < l$ **do**
    $E \leftarrow$ set of all entities in current state.
    if $A' \neq A_k$ **then**
      #IF THERE ARE AXIOMS UNUSED, SAMPLE AN UNUSED AXIOM
      $a \sim \text{Uniform}(A_k \setminus A')$
    else
      #IF ALL AXIOMS ARE USED, SAMPLE A RANDOM AXIOM
      $a \sim \text{Uniform}(A_k)$
    end if
    $A' \leftarrow A' \cup \{a\}$
    $h_{\text{new}} \leftarrow \texttt{NEW}(f, E)$
    $o \leftarrow \texttt{EXT}(f, E)$
    Apply theorem a with operands o; obtain conclusion f'.
    $c \leftarrow c + 1$
    $o_{\text{sub}} \leftarrow \texttt{SUB}(f, f', E)$
    Apply substitution axiom with operands o_{sub}; obtain conclusion f_{new}.
    if $f_{\text{new}} \neq f'$ **then**
      #ONLY INCREMENT COUNTER IF SUBSTITUTION IS NOT REDUNDANT
      $c \leftarrow c + 1$
    end if
    $f \leftarrow f_{\text{new}}$
    $H \leftarrow H \cup h_{\text{new}}$
  end while
  return f, H

t =AdditionCommutativity: 2 inputs: a and b Assumptions: $\emptyset$ Conclusions: $[a + b = b + a]$	$e \sim Uni(E)$ $\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$:, **return** $[LHS(f), e]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(LHS(f')), RHS(f)]$
t =AdditionAssociativity: 3 inputs: a, b and c Assumptions: $\emptyset$ Conclusions: $[a + (b + c) = (a + b) + c]$	$e_1 \sim Uni(E), e_2 \sim Uni(E)$ $\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), e_1, e_2]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(LHS(f')), RHS(f)]$
t =AdditionSimplification: 2 inputs: a and b Assumptions: $[a = 0]$ or $[b = 0]$ Conclusions: $[a + b = b]$ or $[a + b = a]$	$\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$:, **return** $[LHS(f), 0]$ $\texttt{SUB}(f, f', E)$: **return** $[RHS(f'), RHS(f)]$

t =MultiplicationCommutativity: 2 inputs: a and b Assumptions: $\emptyset$ Conclusions: $[a * b = b * a]$	$e \sim Uni(E)$ $\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), e]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(LHS(f')), RHS(f)]$
t =MultiplicationAssociativity: 3 inputs: a, b and c Assumptions: $\emptyset$ Conclusions: $[a * (b * c) = (a * b) * c]$	$e_1 \sim Uni(E), e_2 \sim Uni(E)$ $\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), e_1, e_2]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(LHS(f')), RHS(f)]$
t =MultiplicationSimplification: 2 inputs: a and b Assumptions: $[a = 0]$ or $[b = 0]$ or $[a = 1]$ or $[b = 1]$ Conclusions: $[a * b = 0]$ or $[a * b = 0]$ or $[a * b = b]$ or $[a * b = a]$	$\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$:, **return** $[LHS(f), 1]$ $\texttt{SUB}(f, f', E)$: **return** $[RHS(f'), RHS(f)]$
t =AdditionMultiplicationLeftDistribution: 3 inputs: a, b and c Assumptions: $\emptyset$ Conclusions: $[(b + c) * a = b * a + c * a]$	$e_1 \sim Uni(E), e_2 \sim Uni(E)$ $\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), e_1, e_2]$ $\texttt{SUB}(f, f', E)$: **return** $[RHS(LHS(f')), RHS(f)]$
t =AdditionMultiplicationRightDistribution: 3 inputs: a, b and c Assumptions: $\emptyset$ Conclusions: $[a * (b + c) = a * b + a * c]$	$e_1 \sim Uni(E), e_2 \sim Uni(E)$ $\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), e_1, e_2]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(LHS(f')), RHS(f)]$
t =SquareDefinition: 1 input: a Assumptions: $\emptyset$ Conclusions: $[a^2 = a * a]$	$\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$:, **return** $[LHS(f)]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(LHS(f')), RHS(f)]$
t =SquareGEQZero*: 1 input: a Assumptions: $\emptyset$ Conclusions: $[a^2 \geq 0]$	$e \sim Uni(E)$ $\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$: **return** $[e]$ $\texttt{SUB}(f, f', E)$: **return** $\emptyset$
t =EquivalenceTransitivity: 3 inputs: a, b and c Assumptions: $[a = b,\ b = c]$ Conclusions: $[a = c]$	$e \sim Uni(E)$, new independent variable z, $\texttt{NEW}(f, E)$: **return** $[e + z = LHS(f)]$ $\texttt{EXT}(f, E)$:, **return** $[e + z, LHS(f) RHS(f)]$ $\texttt{SUB}(f, f', E)$: **return** $\emptyset$
t =EquivalenceSymmetry: 2 inputs: a and b Assumptions: $[a = b]$ Conclusions: $[b = a]$	$\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$:, **return** $[LHS(f), RHS(f)]$ $\texttt{SUB}(l, E, G_n, c)$: **return** $\emptyset$
t =PrincipleOfEquality: 4 inputs: a, b, c and d Assumptions: $[a = b,\ c = d]$ Conclusions: $[a + c = b + d]$	$e_1 \sim Uni(E), e_2 \sim Uni(E)$, new independent variable z, $\texttt{NEW}(f, E)$: **return** $[e_1 + z = e_2]$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), RHS(f), e_1 + z, e_2]$ $\texttt{SUB}(f, f', E)$: **return** $\emptyset$
t =EquMoveTerm: 3 inputs: a, b and c Assumptions: $[a + b = c]$ Conclusions: $[a = c + (-b)]$	$e \sim Uni(E)$, new independent variable z, $\texttt{NEW}(f, E)$: **return** $[LHS(f) + z = e]$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), z, e]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(f'), RHS(f)]$
t =IneqMoveTerm: 3 inputs: a, b and c Assumptions: $[a + b \geq c]$ Conclusions: $[a \geq c + (-b)]$	$e \sim Uni(E)$, new independent variable z, $\texttt{NEW}(f, E)$: **return** $[LHS(f) + z \geq e]$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), z, e]$ $\texttt{SUB}(f, f', E)$: **return** $[LHS(f'), RHS(f)]$
t =EquivalenceImpliesDoubleInequality: 2 inputs: a and b	$\texttt{NEW}(f, E)$: **return** $\emptyset$ $\texttt{EXT}(f, E)$:, **return** $[LHS(f), RHS(f)]$

Assumptions: $[a = b]$ Conclusions: $[a \geq b,\ a \leq b]$	$\texttt{SUB}(f, f', E)$: **return** $\emptyset$
$t =$InequalityTransitivity: 3 inputs: a, b and c Assumptions: $[a \geq b,\ b \geq c]$ Conclusions: $[a \geq c]$	$e \sim Uni(E)$, new independent variable z, $\texttt{NEW}(f, E)$: **return** $[e + z \geq LHS(f)]$ $\texttt{EXT}(f, E)$: **return** $[e + z, LHS(f), RHS(f)]$ $\texttt{SUB}(f, f', E)$: **return** $\emptyset$
$t =$FirstPrincipleOfInequality: 4 inputs: a, b, c and d Assumptions: $[a \geq b,\ c \geq d]$ Conclusions: $[a + c \geq b + d]$	$e_1 \sim Uni(E), e_2 \sim Uni(E)$, new independent variable z, $\texttt{NEW}(f, E)$: **return** $[e_1 + z \geq e_2]$ $\texttt{EXT}(f, E)$: **return** $[e_1 + z, e_2, LHS(f), RHS(f)]$ $\texttt{SUB}(f, f', E)$: **return** $\emptyset$
$t =$SecondPrincipleOfInequality: 3 inputs: a, b and c Assumptions: $[a \geq b,\ c \geq 0]$ Conclusions: $[a * c \geq b * c]$	$e \sim Uni(E)$, new independent variable z, $\texttt{NEW}(f, E)$: **return** $[e + z \geq 0]$ $\texttt{EXT}(f, E)$: **return** $[LHS(f), RHS(f), e + z]$ $\texttt{SUB}(f, f', E)$: **return** $\emptyset$
$t =$EquivalenceReflexivity: 1 input: a Assumptions: $\emptyset$ Conclusions: $[a = a]$	NEW, EXT, SUB undefined as the theorem is not in any theorem set T used in the NES synthesis.
$t =$EquivalenceSubstitution: 2 inputs: a and b Assumptions: $[f(a),\ a = b]$ Conclusions: $[f(b)]$	NEW, EXT, SUB undefined as the theorem is not in any theorem set T used in the NES synthesis.

*: This theorem is special as it requires an application of the first principle of inequality after the application of itself.

B Example problems

Equality theorems

Theorem 1
Goal: $((0 \cdot 1) \cdot ((-(a^2)) \cdot c)) = (((-(a^2)) \cdot ((a \cdot a) + (-(a^2)))) \cdot c)$

Theorem 2
Goal: $(((((0 + c) + a) \cdot a) \cdot 1) \cdot (b \cdot (0 + c))) = ((((c \cdot a) + (a \cdot a)) \cdot (0 + c)) \cdot b)$

Theorem 3
Goal: $0 = ((((c + 0) \cdot (a + a)) \cdot (\frac{1}{((c \cdot a)+(c \cdot a))})) + (-(0 + 1)))$

Theorem 4
Premises: $(b + d) = b$
Goal: $(1 + (-((b + b) \cdot (\frac{1}{((b+(b+d)) \cdot 1)})))) = (0 + 0)$

Theorem 5
Premises: $(a + d) = b$
Goal: $1 = (((d \cdot ((a + d) + ((c + (a + d)) + 0))) \cdot ((d \cdot (a + d)) + (d \cdot (c + b)))) \cdot (\frac{1}{((d \cdot ((a+d)+((c+(a+d))+0)))^2)}))$

Theorem 6
Premises: $((b \cdot b) + d) = (b \cdot b)$
Goal: $(0 + ((b \cdot b) + d)) = (((1 \cdot ((b + b) \cdot b)) + (-(((b \cdot b) + (b \cdot b)) \cdot 1))) + (b \cdot b))$

Theorem 7
Goal: $((a \cdot (a + 0)) + ((-(0 + a)) \cdot (a + 0))) = ((a \cdot 0) + (0 \cdot 0))$

Theorem 8
Goal: $(((c \cdot c) + c) \cdot ((c^2) \cdot 1)) = (((c \cdot c) \cdot (0 + (c \cdot c))) + (c \cdot (0 + (c \cdot c))))$

Theorem 9
Goal: $1 = ((((a \cdot c) + ((b \cdot (a \cdot b)) \cdot c)) \cdot (a + (a \cdot c))) \cdot (\frac{1}{((((a+((b \cdot a) \cdot b)) \cdot c) \cdot (a \cdot c))+(((a+((b \cdot a) \cdot b)) \cdot c) \cdot a))}))$

Theorem 10
Goal: $((((b \cdot c) + (c \cdot c)) + (-(0 + ((b + c) \cdot c)))) \cdot (c \cdot c)) = ((c^2) \cdot 0)$

Theorem 11
Goal: $(1 \cdot (b + a)) = ((0 + (a + b)) + 0)$

Theorem 12
Goal: $(((-c) \cdot (-c)) + (((-c) \cdot c) + ((-c) \cdot (-c)))) = (((-c) \cdot (-c)) + (0 \cdot (-c)))$

Theorem 13
Goal: $(((a^2) \cdot (a \cdot (a + 0))) + (a \cdot (a \cdot (a + 0)))) = ((((a^2) \cdot (a^2)) + (a \cdot (a^2))) + 0)$

Theorem 14
Goal: $((((b \cdot 1) \cdot (a \cdot c)) \cdot (b \cdot a)) + (((b \cdot 1) \cdot (a \cdot c)) \cdot (b \cdot a))) = (((((b \cdot a) \cdot c) \cdot (b \cdot a)) + (((b \cdot a) \cdot c) \cdot (b \cdot a))) \cdot 1)$

Theorem 15
Goal: $1 = ((\frac{1}{((\frac{1}{(b+0)}) \cdot b)}) \cdot 1)$

Theorem 16

Goal: $0 = ((0 + (-((a \cdot b) + (-(b \cdot a))))) + (-(0 \cdot 1)))$

Theorem 17
Premises: $(a + d) = c$; $((b + c) + e) = (a + d)$
Goal: $(((b \cdot d) + (b \cdot (b + (a + d)))) + ((b + c) + e)) = ((((b \cdot d) + (b \cdot (b + c))) \cdot 1) + (a + d))$

Theorem 18
Goal: $((((\frac{1}{b}) \cdot b) \cdot b) \cdot 1) = ((b \cdot 1) \cdot 1)$

Theorem 19
Goal: $(((1 \cdot (b \cdot (c + a))) + (b \cdot a)) + 1) = (1 \cdot ((1 \cdot ((b \cdot c) + (b \cdot a))) + ((b \cdot a) + 1)))$

Theorem 20
Premises: $(b + d) = c$; $((1 \cdot a) + e) = a$
Goal: $(((a + (b + d)) \cdot (\frac{1}{((1 \cdot a) + c)})) + ((1 \cdot a) + e)) = ((1 \cdot 1) + a)$

Theorem 21
Goal: $((((c^2) \cdot ((c^2) \cdot c)) + (-(((c \cdot c) \cdot (c^2)) \cdot c))) + (b + b)) = ((1 \cdot ((0 + b) + b)) + (-0))$

Theorem 22
Premises: $(b + d) = (a \cdot b)$
Goal: $(1 \cdot ((((c+c) \cdot (((a \cdot b) \cdot c)+(c+c)))+((c+c) \cdot (c+c)))+(a \cdot b))) = (((((c+c) \cdot ((((a \cdot (b \cdot c))+c)+c)+(c+c)))+(b+d)) \cdot 1)+0)$

Theorem 23
Premises: $((0 \cdot 1) + d) = (1 \cdot 0)$
Goal: $(((((a+(0 \cdot 1)) \cdot (1 \cdot 0))+(-b))+(1 \cdot 0))+(1 \cdot 0)) = (((((a \cdot (1 \cdot 0))+((b+(-b)) \cdot (1 \cdot 0)))+((-b)+(1 \cdot 0)))+((0 \cdot 1)+d))+0)$

Theorem 24
Premises: $(a + d) = (1 + c)$
Goal: $(((((1 \cdot b)+(c \cdot b))+(1+c))^2) \cdot ((1+c) \cdot b)) = ((((((1 \cdot b)+(c \cdot b))+(1+c)) \cdot (((b \cdot (1+(1 \cdot c)))+(a+d)) \cdot 1)) \cdot (1+c)) \cdot b)$

Theorem 25
Premises: $(a + d) = (b \cdot 1)$
Goal: $0 = ((b + (a + d)) + (-((b \cdot 1) + (b \cdot 1))))$

Theorem 26
Premises: $(c + d) = a$
Goal: $(0 + ((((a + a) \cdot 1) + a) \cdot 1)) = (1 \cdot ((1 \cdot ((a + (c + d)) + 0)) + (1 \cdot a)))$

Theorem 27
Premises: $(c + d) = (b + c)$
Goal: $(1 \cdot ((((b + c) \cdot c) + (b \cdot (b + c))) + (c + d))) = ((((b + c)^2) + (b + c)) \cdot 1)$

Theorem 28
Premises: $((1 \cdot b) + d) = b$
Goal: $(((((1 \cdot b) + b) \cdot (a \cdot 1)) \cdot (((b + ((1 \cdot b) + d)) \cdot a) \cdot 1)) + 0) = ((((b + (1 \cdot b)) \cdot (a \cdot 1))^2) \cdot 1)$

Theorem 29
Goal: $(((b \cdot 1) + 0) \cdot (1 \cdot 0)) = (((b \cdot 1) \cdot ((-(0 + b)) + (1 \cdot b))) + (0 \cdot ((-(0 + b)) + (1 \cdot b))))$

Theorem 30

Goal: $(1 \cdot 1) = (((((a \cdot (c + c)) + 0) \cdot (b \cdot (c + c))) \cdot (\frac{1}{((((a \cdot c) + (a \cdot c)) \cdot b) \cdot (c + c))})) + 0)$

Theorem 31
Goal: $((1 \cdot (b \cdot b)) \cdot b) = (1 \cdot (0 + (((0 + b) \cdot b) \cdot b)))$

Theorem 32
Goal: $(((c \cdot (c \cdot 1)) + 0) \cdot 1) = (((c \cdot c) + 0) \cdot 1)$

Theorem 33
Goal: $1 = (1 \cdot (\frac{1}{((1+0) \cdot (\frac{1}{((b \cdot (\frac{1}{b})) + 0)}))}))$

Theorem 34
Goal: $(((((((c + a) \cdot a) \cdot (c + a)) \cdot c) \cdot (a + c)) \cdot (c + a)) \cdot (c + a)) = (((((((a + c) \cdot (c + a)) \cdot a) \cdot c) \cdot (a + c)) \cdot (c + a)) \cdot (c + a))$

Theorem 35
Goal: $0 = ((-(1 \cdot 0)) + ((-(c + c)) + ((1 \cdot c) + c)))$

Theorem 36
Goal: $1 = (1 \cdot (\frac{1}{(a \cdot (\frac{1}{(((a+c)+a)+(-(c+a)))}))}))$

Theorem 37
Premises: $(a + d) = a$; $((\frac{1}{c}) + e) = b$
Goal: $(((1 \cdot (1 \cdot (\frac{1}{(c \cdot (\frac{1}{c}))}))) + a) + b) = (1 \cdot (((1 \cdot 1) + (a + d)) + ((\frac{1}{c}) + e)))$

Theorem 38
Goal: $0 = ((b \cdot (b + (-b))) + (-(((0 + 0) \cdot b) + 0)))$

Theorem 39
Goal: $(((1 \cdot c) + (-(1 \cdot (c \cdot 1)))) \cdot 1) = ((0 \cdot 1) \cdot 1)$

Theorem 40
Goal: $((a + b) \cdot (1 \cdot ((b \cdot c) + (c \cdot c)))) = ((a \cdot ((c \cdot c) + (b \cdot c))) + (b \cdot ((c \cdot c) + (b \cdot c))))$

Theorem 41
Goal: $(0 + ((0 + ((c + c) \cdot c)) \cdot (a \cdot b))) = (0 + ((((c \cdot c) \cdot a) + ((c \cdot c) \cdot a)) \cdot b))$

Theorem 42
Premises: $(0 + d) = 1$
Goal: $((((1 \cdot 0) + (a + (a \cdot 1))) + 0) + d) = (((((1 \cdot a) + (-(a \cdot 1))) + a) + (a \cdot 1)) + 1)$

Theorem 43
Premises: $(b + d) = 0$
Goal: $0 = ((((((0 + b) \cdot 0) + ((0 + b) \cdot b)) \cdot 1) + 0) + (-((((b \cdot 0) + (b \cdot b)) + (b + d)) \cdot 1)))$

Theorem 44
Goal: $((0 + c) \cdot ((-c) + (((c \cdot 1) + 0) + (-c)))) = (((0 + c) \cdot (-c)) + ((0 + c) \cdot 0))$

Theorem 45
Goal: $0 = (0 + (-(((0 \cdot 0) + (a \cdot 0)) + (-(((((((a \cdot b) + (a \cdot b)) + ((b + b) + b)) + (-(((a \cdot (b + b)) + (b + b)) + b))) + a) \cdot 0) \cdot 1)))))$

Theorem 46

Premises: $((a+b)+d)=(a+b)$; $(b+e)=a$
Goal: $(a \cdot a)=(1 \cdot (a \cdot a))$

Theorem 47
Premises: $(c+d)=c$
Goal: $((b \cdot (1+0))+(b \cdot (c+d)))=(0+(b \cdot ((0+(1 \cdot (\frac{1}{((b+(c+d)) \cdot (\frac{1}{(b+c)}))})))+(c+d))))$

Theorem 48
Goal: $((b+((((a+a) \cdot 1) \cdot a)+0)) \cdot a)=((b \cdot a)+((((a \cdot a) \cdot 1)+(a \cdot (a \cdot 1))) \cdot a))$

Theorem 49
Goal: $(((1+b) \cdot (((a \cdot ((c \cdot 1)+(c^2)))+1)+b))+((1+b) \cdot (((a \cdot ((c \cdot 1)+(c^2)))+1)+b)))=((((1+b)+(1+b)) \cdot ((((a \cdot (c \cdot 1))+(a \cdot (c \cdot (c \cdot 1))))+(1+b)) \cdot 1)) \cdot 1)$

Theorem 50
Goal: $0=(((((0+(((c \cdot c)+(c \cdot c))+((c+c)+(c \cdot c))))+0)+c) \cdot a)+(-(((0+((((c+c) \cdot c)+(c+c))+(c \cdot c))) \cdot a)+(c \cdot a))))$

Inequality theorems

Theorem 1
Premises: $(1+d) \geq 0$; $(b+e) \geq 0$
Goal: $((((1+1) \cdot (a \cdot (\frac{1}{a}))) \cdot (1+d))+(b+e)) \geq ((((1 \cdot 1)+(1 \cdot 1)) \cdot (1+d))+0)$

Theorem 2
Goal: $(b^2) \geq (0+(b \cdot (1 \cdot b)))$

Theorem 3
Premises: $((c+0)+d) \geq 0$; $(d+e) \geq b$
Goal: $((c \cdot ((c+0)+d))+(d+e)) \geq (((0+c) \cdot ((c+0)+d))+b)$

Theorem 4
Goal: $(b+0) \geq ((((0+b)+c)+c)+(-(c+c)))$

Theorem 5
Premises: $(1+d) \geq 0$
Goal: $((((c \cdot c)+c)+a) \cdot (1+d)) \geq ((((c^2)+(c+a)) \cdot 1) \cdot (1+d))$

Theorem 6
Premises: $(b+d)=b$
Goal: $1 \geq ((((a+b)+(-(b+d))) \cdot ((a+b)+b)) \cdot (\frac{1}{((a \cdot (a+b))+(a \cdot b))}))$

Theorem 7
Premises: $((0+a)+d)=0$
Goal: $(((0+a) \cdot a)+((0+a)+d)) \geq ((a^2)+0)$

Theorem 8
Premises: $(b+d)=a$
Goal: $((c \cdot b)+(b \cdot b)) \geq (1 \cdot ((((c+a)+(-(b+d)))+b) \cdot b))$

Theorem 9
Goal: $(1 \cdot ((((b \cdot (\frac{1}{b})) \cdot a) \cdot (1 \cdot 1)) + a)) \geq ((1 \cdot ((1 \cdot 1) \cdot (a \cdot (1 \cdot 1)))) + (1 \cdot a))$

Theorem 10
Premises: $(c + d) \geq 0$
Goal: $(b \cdot (c + d)) \geq ((((b + b) + 0) + (-b)) \cdot (c + d))$

Theorem 11
Goal: $(((b + 0) + (b + c)) + 0) \geq (((b + b) + c) + 0)$

Theorem 12
Goal: $((c \cdot (c + 0)) + 0) \geq ((c^2) + 0)$

Theorem 13
Goal: $(1 \cdot (b \cdot 1)) \geq ((1 \cdot b) \cdot 1)$

Theorem 14
Goal: $1 \geq ((((b \cdot (\frac{1}{b})) + (\frac{1}{b})) + 1) \cdot (\frac{1}{((1+(\frac{1}{b}))+1)}))$

Theorem 15
Goal: $1 \geq ((\frac{1}{((c \cdot a) \cdot (\frac{1}{(a \cdot c)}))}) \cdot 1)$

Theorem 16
Goal: $((c \cdot (a \cdot a)) + (((a \cdot a) + (c \cdot a)) \cdot (a \cdot a))) \geq (0 + ((c + (0 + ((a + c) \cdot a))) \cdot (a \cdot a)))$

Theorem 17
Goal: $(((c \cdot b) + a) \cdot ((c \cdot b) + (c \cdot b))) \geq ((a \cdot ((c \cdot b) + (c \cdot b))) + ((c \cdot b) \cdot ((c \cdot b) + (c \cdot b))))$

Theorem 18
Goal: $((a \cdot b) \cdot 1) \geq ((((a \cdot 1) \cdot b) \cdot 1) \cdot 1)$

Theorem 19
Goal: $a \geq ((a + c) + (-c))$

Theorem 20
Goal: $((c \cdot b) \cdot b) \geq (b \cdot (b \cdot c))$

Theorem 21
Premises: $(a + d) = a$; $((a + d) + e) \geq 0$; $(b + f) \geq (0 \cdot 0)$
Goal: $((((((c \cdot 0) + (0 \cdot 0)) + (a + d)) \cdot ((0 + ((c + 0) \cdot (a + (-a)))) + a)) \cdot ((a + d) + e)) + (b + f)) \geq ((0 \cdot ((a + d) + e)) + (0 \cdot 0))$

Theorem 22
Premises: $(c + d) \geq 0$; $((0 + 0) + e) \geq (0 + 0)$
Goal: $((((((0 + (c + (-c))) \cdot (-c)) \cdot (\frac{1}{((0 \cdot (-c)) + (0 \cdot (-c)))})) \cdot (0 + 1)) \cdot (c + d)) + ((0 + 0) + e)) \geq ((0 \cdot (c + d)) + (0 + 0))$

Theorem 23
Premises: $((a^2) + d) \geq 0$
Goal: $((((a \cdot a) + c) \cdot (0 + (1 \cdot (a \cdot a)))) \cdot ((a^2) + d)) \geq ((((a \cdot a) \cdot ((a^2) + 0)) + (c \cdot ((a^2) + 0))) \cdot ((a^2) + d))$

Theorem 24
Premises: $(c + d) = c$; $((0 + a) + e) \geq a$

Goal: $((((a+b)\cdot(((a+(-a))+(a+b))+(c+d)))\cdot(((((0+a)+b)+c)\cdot(a+b))\cdot 1))+((0+a)+e))\geq(0+a)$

Theorem 25
Goal: $1\geq((a\cdot(c+b))\cdot(\frac{1}{((a\cdot c)+(a\cdot b))}))$

Theorem 26
Premises: $(a+d)\geq b$
Goal: $((0\cdot((((((a+c)+a)\cdot(a\cdot c))\cdot(a\cdot c))+((a\cdot c)\cdot(a\cdot c)))+(-((((((a+(c+a))\cdot a)\cdot c)+(a\cdot c))\cdot(a\cdot c))+0))))+(a+d))\geq(0+b)$

Theorem 27
Premises: $((c\cdot b)+d)=(b\cdot b)$; $((b\cdot b)+e)\geq a$
Goal: $((((b+b)+(b+b))\cdot((((c\cdot(b\cdot b))+b)+b)+(b^2)))+((b\cdot b)+e))\geq((((b+b)\cdot((((c\cdot b)\cdot b)+(b+b))+((c\cdot b)+d)))+((b+b)\cdot((((c\cdot b)\cdot b)+(b+b))+((c\cdot b)+d))))+a)$

Theorem 28
Premises: $((b\cdot 0)+d)\geq c$
Goal: $((((b+(((0+c)+(0+c))+0))\cdot 0)\cdot((b\cdot 0)+(((0+c)+(0+c))\cdot 0)))+((b\cdot 0)+d))\geq(0+c)$

Theorem 29
Premises: $(a+d)\geq 0$
Goal: $((0\cdot((((c\cdot c)+(c\cdot 0))\cdot a)+(-(((c+0)\cdot((c+0)\cdot a))\cdot 1))))+(a+d))\geq(0+0)$

Theorem 30
Premises: $(a+d)\geq c$
Goal: $(((b\cdot(b\cdot 1))+(b\cdot c))+(a+d))\geq((0+(b\cdot((b\cdot 1)+c)))+c)$

Theorem 31
Goal: $(0+(0+(c+b)))\geq(0+((b+c)+0))$

Theorem 32
Goal: $(a+(a+0))\geq((((0+a)+0)+a)+0)$

Theorem 33
Premises: $((c+c)+d)\geq a$; $(d+e)\geq 0$; $((c+c)+f)\geq(0+a)$; $(b+g)\geq 0$
Goal: $(((((((c+c)+(c+c))\cdot((c+c)+(c+c)))+((c+c)+d))+(d+e))+((c+c)+f))+(b+g))\geq((((0+a)+0)+(0+a))+0)$

Theorem 34
Goal: $(((0+b)+c)+a)\geq(0+(0+(b+(c+a))))$

Theorem 35
Premises: $(a+d)\geq 0$; $(a+e)\geq(c\cdot c)$; $(e+f)\geq 0$; $(c+g)\geq 0$; $(c+h)\geq(c+g)$; $(c+i)\geq 0$
Goal: $(((((((c\cdot c)\cdot(a+d))+(a+e))\cdot(e+f))\cdot(c+g))+(c+h))\cdot(c+i))\geq((((((0\cdot(a+d))+(c\cdot c))\cdot(e+f))\cdot(c+g))+(c+g))\cdot(c+i))$

Theorem 36
Goal: $(1\cdot(1\cdot(1\cdot a)))\geq(1\cdot((a+0)+0))$

Theorem 37
Premises: $(b+d)\geq b$; $((c+b)+e)\geq c$; $(b+f)\geq a$; $(e+g)\geq(b+f)$
Goal: $(((c+(b+d))+(b+f))+(e+g))\geq(((((c+b)+c)+(-((c+b)+e)))+a)+(b+f))$

Theorem 38
Goal: $((a+(((b+c)\cdot(b+c))+((c+b)\cdot b)))\cdot((c+b)+(c+b))) \geq ((((((b+c)\cdot(c+b))+((b+c)\cdot b))+a)\cdot(c+b))+(((((b+c)\cdot(c+b))+((b+c)\cdot b))+a)\cdot(c+b)))$

Theorem 39
Premises: $(c+d)=b$; $((c+b)+e)=(c+d)$; $(a+f)\geq 0$; $(0+g)\geq 0$; $(g+h)\geq 0$; $(d+i)\geq 0$
Goal: $((((((c+(c+d))+((c+b)+e))\cdot(a+f))\cdot(0+g))\cdot(g+h))\cdot(d+i)) \geq ((((((c+b)+(c+d))\cdot(a+f))\cdot(0+g))\cdot(g+h))\cdot(d+i))$

Theorem 40
Goal: $((((c+a)\cdot b)\cdot b)+(a+c)) \geq ((a+c)+(((a+c)\cdot b)\cdot b))$

Theorem 41
Goal: $(((c+b)+(a+(c+b)))\cdot(\frac{1}{((((1\cdot c)+b)+a)+(c+b))})) \geq (1\cdot 1)$

Theorem 42
Premises: $(c+d)=b$
Goal: $(((((c\cdot b)+(c^2))\cdot((b+c)\cdot(c\cdot b)))+(c+d))\cdot(((((c\cdot(b+c))\cdot(b+c))\cdot c)\cdot b)+b)) \geq (((((c\cdot b)+(c^2))\cdot((b+c)\cdot(c\cdot b)))+(c+d))^2)$

Theorem 43
Premises: $(a+d)=b$; $(d+e)=a$; $(c+f)\geq 0$; $((b+b)+g)\geq 0$
Goal: $((1\cdot(c+f))\cdot((b+b)+g)) \geq (((((b+b)+a)\cdot(\frac{1}{(0+((b+(a+d))+(d+e)))}))\cdot(c+f))\cdot((b+b)+g))$

Theorem 44
Goal: $(((((a\cdot 1)\cdot a)\cdot 1)\cdot b)+(((a\cdot 1)\cdot(a\cdot 1))\cdot(a\cdot a))) \geq (1\cdot((((a\cdot a)\cdot 1)\cdot b)+(((a\cdot a)\cdot 1)\cdot(a\cdot a))))$

Theorem 45
Premises: $((c+0)+d)\geq b$; $(1+e)\geq a$
Goal: $((0+((c+0)+d))+(1+e)) \geq (((0+(-((c\cdot 1)+(-(c+0)))))+b)+a)$

Theorem 46
Premises: $(c+d)\geq(a\cdot c)$
Goal: $(((1\cdot(1\cdot(a\cdot(a\cdot c))))\cdot((1\cdot((a\cdot a)\cdot c))+0))+(c+d)) \geq (0+(a\cdot c))$

Theorem 47
Premises: $(c+d)\geq c$
Goal: $((c\cdot(0+c))^2) \geq (((0+((c\cdot(0+c))\cdot(c^2)))+c)+(-(c+d)))$

Theorem 48
Premises: $(a+d)=b$
Goal: $(1\cdot((b+b)+(-(1\cdot(b+(a+d)))))) \geq (1\cdot(0\cdot 1))$

Theorem 49
Premises: $((c\cdot b)+d)=a$; $((c\cdot b)+e)\geq b$
Goal: $(((b\cdot b)\cdot(a\cdot(c\cdot b)))+((c\cdot b)+e)) \geq ((((b\cdot b)\cdot a)\cdot(c\cdot b))+b)$

Theorem 50
Goal: $(((a+c)\cdot(c+a))+((a\cdot(c+a))+((c\cdot c)+(c\cdot a)))) \geq (((a+c)\cdot((c+a)+(c+a)))\cdot 1)$