NLIR: Natural Language Intermediate Representation for Mechanized Theorem Proving

Anonymous Author(s) Affiliation Address email

Abstract

1	Formal theorem proving is challenging for humans as well as for machines. Thanks
2	to recent advances in LLM capabilities, we believe natural language can serve as a
3	universal interface for reasoning about formal proofs. In this paper, 1) we introduce
4	<i>Pétanque</i> , a new lightweight environment to interact with the Coq theorem prover;
5	2) we present two interactive proof protocols leveraging natural language as an
6	intermediate representation for designing proof steps; 3) we implement beam
7	search over these interaction protocols, using natural language to rerank proof
8	candidates; and 4) we use Pétanque to benchmark our search algorithms. Using
9	our method with GPT-40 we can successfully synthesize proofs for 46% of the
10	Logical Foundation series and for 50% of the first 100/260 lemmas from the newly
11	published Busy Beaver proofs. ¹

12 1 Introduction

The general knowledge and reasoning abilities of frontier large language models (LLMs) makes 13 them practical as a backbone for building agents able to interact with theorem provers. These agents 14 should iteratively build proofs with help from proof engine feedback. While previous work (e.g. Yang 15 et al. [2023]) used a costly data collection procedure to finetune modestly sized language models, 16 17 we believe that reasoning in natural language before outputting tactics will lead to better and more interpretable results. Recently, Thakur et al. [2024] showed promising preliminary results by using 18 19 GPT-4 as an agent proposing tactics inside a backtracking search and using rich feedback from the proof environment. 20

In this work, we develop infrastructure to allow communication between a GPT-4o-based agent 21 and the Coq proof environment [The Coq Development Team, 2024]. Our key idea is to rely on 22 natural language as much as possible when generating proofs. Using natural language leverages the 23 strength of LLMs, and allows us to use chain-of-thought [Wei et al., 2022] by asking for an informal 24 25 mathematical proof before generating the formal proof, making it more intuitive and comprehensible compared to purely automatic formal techniques. Additionally, partial proofs expressed in natural 26 language are easier for humans to understand, adapt, or reuse, allowing for greater flexibility and 27 collaboration between machine-generated suggestions and human mathematicians. 28

We present the following contributions: 1) *Pétanque*: A new fast and lightweight environment to interact with the Coq theorem prover. 2) Two interactive proof protocols both leveraging natural language reasoning: tactic-by-tactic proof construction, and hierarchical proof templating. 3) We couple both protocols with standard search algorithms leveraging feedback from the ITP and using

atural language to rerank proof candidates. 4) We evaluate this agent on a new dataset of textbook

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

¹https://github.com/ccz181078/Coq-BB5



exercises and intermediate theorems from the recent Busy Beaver proof formalized in Coq of BB(4) = 107, [ccz181078, 2024].

³⁶ 2 Pétanque: a lightweight interactive environment for Coq

A common difficulty when interacting with interactive proof assistants in the context of machine 37 learning is inadequate tooling, see for example [Reichel et al., 2023]. Following existing work 38 [Gallego Arias et al., 2016, Gallego Arias, 2019, Yang and Deng, 2019, Sanchez-Stern et al., 2020], 39 we have built a new environment for machine to machine interaction for the Coq proof assistant, 40 particularly tailored for interactive, high-throughput, low-latency learning applications. Pétanque 41 42 is based on Flèche [Gallego Arias, 2024], a new document manager for Coq. We extend Flèche by 43 enabling Pétanque to access the Coq proof engine directly without requiring edits in the associated document. This makes our environment fast and lightweight. A Python interface, pytanque, provides 44 easy access to the API. 45

46 3 Proof interaction protocols

In this section, we present two approaches leveraging LLMs' ability to reason in natural language in 47 order to find a formal proof with the help of a proof assistant. Tactic-by-tactic proof construction 48 mimics the typical behavior of a standard Coq user: given the current goals, the agent generates 49 one or several tactics that updates the goals and repeats this process until the proof is complete. By 50 contrast, hierarchical proof templating tries to generate full proofs directly. Failed tactics are then 51 replaced with *holes* to obtain a proof *template*. The agent then repeats the process of filling each hole 52 until the proof is complete. Our approach's originality is that although both protocols' inputs (goals) 53 and outputs (tactics) are in Coq code, the agent internally uses natural language as an intermediate 54 representation to analyze the input and guide the code generation. 55

56 3.1 Tactic-by-tactic proof construction

An overview of the tactic-by-tactic proof construction agent is presented in Figure 1. Given a Coq theorem, the agent first uses natural language to describe the goal and explain how to continue the proof (chain-of-thought). The last step synthesizes the corresponding Coq tactics. For instance, in Figure 1, the goal is to prove that addition over natural numbers is commutative. The agent decides to try a proof by induction and correctly synthesizes a sequence of two tactics: **intros** n m. introduces two variables n and m of type nat (natural number), and **induction** n. starts an induction over n. The tactics are sent to the Pétanque environment, which parses and executes each tactic to update the current goal. A textual representation of the new goal is then fed back to the agent allowing it

the current goal. A textual representation of the new goal is then fed back to the agent, allowing it to progress further in the proof. If the execution returns an error, the current goal does not change, but we augment the prompt with the failed tactics and ask the LLM to try something else for the next attempt. For instance, in Figure 1, both tactics succeed and generate two new subgoals: the base case (for n=0, prove m + 0 = 0 + m) and the induction case (given the induction hypothesis



Figure 2: Hierarchical proof templating.

⁶⁹ IHn: n + m = m + n, prove (n + 1) + m = m + (n + 1). The textual representation of a goal ⁷⁰ uses the the symbol \vdash to separate hypotheses from the conclusion, and S n denotes n + 1.

Model Interface. In early experiments, we observed that conversation-style reasoning often diverges: after a few rounds, the output makes very little sense, and the agent never recovers. Following Yang et al. [2024] – and similarly to Thakur et al. [2024] – we use a synthetic interface to summarize at each goal the global objective (initial theorem), the current goal (in the middle of a proof), and failed attempts to solve the same goal.

attempts to solve the sume gour.

76 **3.2** Hierarchical proof templating

An example execution of the hierarchical proof templating agent is presented in Figure 2. The agent pipeline is similar to the tactic-by-tactic method, but instead of focusing only on the next step, the agent generates a complete proof in natural language, before translating the proof in Coq syntax. For instance, in Figure 2, the agent uses the **inversion** tactics on the hypothesis H which generate two subgoals with a simpler hypothesis H0, and then tries to solve each subgoals using this H0 hypothesis.

Then, rather than simply checking the proof, the Pétanque environment repairs it, by replacing failed 82 tactics with *holes* which admits and closes the current subgoal, removing subsequent tactics until 83 the focus moves to the next subgoal. Pétanque then checks that the resulting *template* is correct, i.e., 84 assuming a valid proof for each holes, the proof is complete. A textual representation of each holes 85 is then fed back to the agent which repeat the process to fill the holes one by one. For instance, in 86 Figure 2, apply H0 fails on both subgoals. The agent then repeats the process for each holes, using 87 focused fine-grain reasoning to prove the corresponding subgoal. The proof is complete when there 88 are no more holes. 89

90 4 Proof search

91 We combine our interactive protocol with the classic

- beam search algorithm. Inspired by Yao et al. [2023],
- ⁹³ we use the LLM to rank and sort the proposals at each
- step of the search.
- 95 A simplified version of the code is presented on the
- ⁹⁶ right. At each step, the agent.generate method
- 97 generates multiple possible steps (tactics or proofs).
- ⁹⁸ Each step is then validated with the petangue.step
- ⁹⁹ method. and the state and the current proof of all the

Init s = petanque.start(thm) beam = [(s,[])] # (state, proofs) pairs for step in range(n_steps) # Generate candidates candidates = [] for (s, p) in beam: # Try multiple actions for each state for a in agent.generate(s, n_actions) sa = petanque.step(s, a) pa = p + [a] # Proof found! if petanque.proof_finished(sa): return pa else: candidates = candidates + [(sa, pa)] # Rank candidates beam = agent.sort(candidates)[:beam_size]

def beam_search(n_steps, n_actions, beam_size):

resulting candidates is stored. The agent. sort method then calls the LLM to discuss, compare and finally rank the candidates for the next step.

102 5 Evaluation

Logical Foundations exercises: We extracted the exercises of *Logical Foundations* [Pierce et al.,
 2024], the first volume of the *Software Foundation* textbooks series that is widely used to introduce
 Coq. We extracted 179 exercises. Given the popularity of this textbook the risk of data leak is high.
 We filtered out 66 "easy" exercises that are solved with one shot prompting (see ?? in ??). This
 dataset thus comprises 113 exercises.

BB(4) lemmas: To avoid data leak issues, we extracted the 260 lemmas from the recent proof of BB(4) = 107 [ccz181078, 2024]. The repository was created in April 2024, long after the knowledge cutoff date of GPT-40 (October 2023). To provide the necessary context for the proof, for each lemma we augment the prompt with all the preceding definitions and lemmas.

Evaluation. The results are presented in the following table. The gray number in the *template* column indicates the number of proofs that were correct at the first try (no holes). On both dataset, we observe that the templating agent coupled with beam search

		Logi	cal Foundation	BB(4)					
	tac	tics	temj	plate	template				
	naive	beam	naive	beam	naive	beam			
% success	30.1	40.7	(16.8) 29.2	(23.0) 46.0	(24.0) 35.0	(40.0) 50.0			

We use Coq 8.19.2 and GPT-40 (Sept. 2024) for all experiments. We observe that the template agent coupled with beam search (n_steps=10, n_actions=4, beam_size=3) outperforms the tactic agent on the Logical Foundation benchmark. To limit the costs of our experiments, we only run the template agent on the first 100 Lemmas of the BB(4) benchmark. For the template agent, the gray numbers indicate the proportion of proofs that are correct at the first try (no holes).

120 6 Related work and conclusion

LLMs and theorem provers Automatic theorem-proving is a longstanding challenge in computer 121 science Newell et al. [1957]. Recent work has used neural models based on autoregressive language 122 model that generate a proof tactic by tactic. Most works use finetuned LLMs [Polu and Sutskever, 123 2020, Han et al., 2021, Wu et al., 2022, Yang et al., 2023, First et al., 2023], trained on (goal, tactic) 124 pairs obtained from intermediate steps of existing proofs. On the other hand, Lample et al. [2022] 125 uses online training, progressively collecting more data. Closest to our work, Thakur et al. [2024] 126 build a tactic-by-tactic LLM agent based on GPT-4 and also use an interface to summarize past 127 128 interactions. They, however, do not use proof repair or beam search. Other work close to ours is Wang et al. [2024], who use proof repair over hierarchical proofs in Isabelle, coupled with best-first 129 search. Contrary to us, they use fine-tuned models and no chain-of-thought. 130

131 **Reasoning in LLMs** This work is also related to recent investigations on the reasoning abilities of LLMs [Plaat et al., 2024]. Chain-of-Thought (CoT) prompting [Wei et al., 2022] was shown 132 to improve LLM's answers; subsequent work found that these reasoning abilities could be elicited 133 zero-shot [Kojima et al., 2022]. Further work interleaved CoT with decision-making [Yao et al., 134 2022], added search and complex control flow to reasoning [Chen et al., 2022, Yao et al., 2023, Besta 135 et al., 2024], incorporated refinement and feedback [Madaan et al., 2024, Shinn et al., 2024], and 136 learned to generate novel reasoning traces that proved beneficial for further training [Zelikman et al., 137 2022, 2024]. Like our work, many of these methods - especially the ones using search and refinement 138 make use of LLM-based scoring or ranking functions [Zheng et al., 2023]. 139

Conclusion In this work, we have presented a new agent for building proofs leveraging chain of thought as an intermediate representation, and generating proofs by outputting step-by-step tactics or hierarchical proof templates. We couple this with beam search and natural language reranking and obtain good performance on a new evaluation set built with the help of our novel proof environment, *Pétanque*. Future work could investigate how one could use reinforcement learning to obtain better reasoning and performance with smaller models [OpenAI, 2024].

146 **References**

- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi,
 Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts:
- Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference*
- *on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.
- 151 ccz181078. https://github.com/ccz181078/Coq-BB5/tree/main, 2024.
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompt ing: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.
- Emily First, Markus N. Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-proof generation and
 repair with large language models. *CoRR*, abs/2303.04910, 2023.
- Emilio Jesús Gallego Arias, Benoît Pin, and Pierre Jouvelot. jscoq: Towards hybrid theorem proving interfaces. In Serge Autexier and Pedro Quaresma, editors, *Proceedings of the 12th Workshop on User Interfaces for Theorem Provers, UITP 2016, Coimbra, Portugal, 2nd July*
- 2016, volume 239 of EPTCS, pages 15–27, 2016. doi: 10.4204/EPTCS.239.2. URL https:
 //doi.org/10.4204/EPTCS.239.2.
- Emilio Jesús Gallego Arias. SerAPI: Machine-friendly, data-centric serialization for Coq. preprint,
 01 2019. URL https://github.com/ejgallego/coq-serapi/.
- Emilio Jesús Gallego Arias. Flèche: Incremental validation for hybrid formal documents. under
 revision, 2024.
- Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W Ayers, and Stanislas Polu. Proof artifact
 co-training for theorem proving with language models. *arXiv preprint arXiv:2102.06203*, 2021.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large
 language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:
 22199–22213, 2022.
- Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat,
 Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. Hypertree proof search for neural theorem
 proving. Advances in neural information processing systems, 35:26337–26349, 2022.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri
 Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement
 with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.
- Allen Newell, John Clifford Shaw, and Herbert A Simon. Empirical explorations of the logic theory
 machine: a case study in heuristic. In *Papers presented at the February 26-28, 1957, western joint computer conference: Techniques for reliability*, pages 218–230, 1957.
- 180 OpenAI. Learning to Reason with LLMs. https://openai.com/o1/, 2024.
- Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael
 Greenberg, Cătălin Hriţcu, Vilhelm Sjöberg, and Brent Yorgey. *Logical Foundations*, volume 1 of
 Software Foundations. Electronic textbook, 2024. Version 6.7, http://softwarefoundations.
 cis.upenn.edu.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back.
 Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*, 2024.
- 187 Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving.
 188 *CoRR*, abs/2009.03393, 2020.
- Tom Reichel, R. Wesley Henderson, Andrew Touchet, Andrew Gardner, and Talia Ringer. Proof repair
 infrastructure for supervised models: Building a large proof repair dataset. In Adam Naumowicz
 and René Thiemann, editors, *14th International Conference on Interactive Theorem Proving, ITP 2023, July 31 to August 4, 2023, Białystok, Poland*, volume 268 of *LIPIcs*, pages 26:1–26:20.
 Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPICS.ITP.2023.26.
 URL https://doi.org/10.4230/LIPIcs.ITP.2023.26.

- Alex Sanchez-Stern, Yousef Alhessi, Lawrence K. Saul, and Sorin Lerner. Generating correctness
 proofs with neural networks. In *MAPL@PLDI*, 2020.
- ¹⁹⁷ Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion:
- Language agents with verbal reinforcement learning. Advances in Neural Information Processing
 Systems, 36, 2024.
- Amitayush Thakur, George D. Tsoukalas, Yeming Wen, Jimmy Xin, and Swarat Chaudhuri. An
 in-context learning agent for formal theorem-proving. In *COLM*, 2024.
- The Coq Development Team. The Coq reference manual release 8.19.0. https://coq.inria.fr/ doc/V8.19.0/refman, 2024.
- Haiming Wang, Huajian Xin, Zhengying Liu, Wenda Li, Yinya Huang, Jianqiao Lu, Zhicheng Yang,
 Jing Tang, Jian Yin, Zhenguo Li, and Xiaodan Liang. Proving theorems recursively. *CoRR*,
 abs/2405.14414, 2024.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi,
 Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language
 models. In *NeurIPS*, 2022.
- Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and
 Christian Szegedy. Autoformalization with large language models. In *NeurIPS*, 2022.
- John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan,
 and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering.
 CoRR, abs/2405.15793, 2024.
- Kaiyu Yang and Jia Deng. Learning to prove theorems via interacting with proof assistants. In *ICML*, 2019.
- Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil,
 Ryan J. Prenger, and Animashree Anandkumar. Leandojo: Theorem proving with retrievalaugmented language models. In *NeurIPS*, 2023.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao.
 React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan.
 Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.
- Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang,
 Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica.
- Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *NeurIPS*, 2023.

233 A Prompts

234 A.1 Tactic-by-tactic proof construction prompt example

235 Instructions

You are an analytical and helpful assistant proficient in mathematics as well as in the use of the Coq theorem prover and programming language. You will be provided with a Coq/math-comp theorem and your task is to prove it. This will happen in interaction with a Coq proof engine which will execute the proof steps you give it, one at a time, and provide feedback. This is the important information about this task:

241 Coq engine interface

242 You will be provided with:

- This information prompt;
- The theorem to prove;
- Successful proof steps until now (current proof);
- Unsucessful proof step attempts with the current goal(s), if any; you know these techniques didn't work, so try avoid reusing them;
- The current goal;

249 Interaction

- 250 Your goal is to write proof steps interactively until you manage to find a complete proof for the
- proposed theorem. You will be able to interact with the proof engine by issuing the following commands:
- 253 **Step** : Passes the string that is given after it to the Coq proof engine. Example usage:

/step intros.

You can use several steps in each interaction, but try to be concise and advance one step at a time, especially if you've been getting errors.

256 **Theorem and proof information**

257 You have interacted 2 times with the engine.

258 **Theorem**

259 Here is the theorem to prove:

forall f : nat -> nat,
(forall n : nat, n = f (f n)) -> forall n1 n2 : nat, f n1 = f n2 -> n1 = n2

260 **Proof**

²⁶¹ Here are the proof steps until now:

intros f H n1 n2 H0.

262 **Previous unsuccessful steps**

²⁶³ Here are the previous unsuccessful proof step attempts. These have all been tried before with the

current goal(s). DOT NOT TRY ANY OF THESE STEPS, as you know they don't work. You should
 try something different.

rewrite H0.

266 Current goal(s)

```
f : nat -> nat
H : forall n : nat, n = f (f n)
n1 : nat
n2 : nat
|- a2 f n1 = f n2 -> n1 = n2
```

267 A.2 Hierarchical proof templating prompt example

Your task is to complete a proof using the Coq proof assistant. For each theorem, I will give you the goal to prove in Coq syntax.

270 Here are a few examples:

```
<example>
<goal>
n, m, p : nat
|- nat, n + (m + p) = m + (n + p)
</goal>
<proof>
rewrite Nat.add_assoc. rewrite Nat.add_assoc.
assert (n + m = m + n) as H by apply Nat.add_comm.
rewrite H. reflexivity.
</proof>
</example>
```

271 [...]

²⁷² Think before you write the proof in <thinking> tags. First explain the goal. Then describe the proof

step by step. Finally write the corresponding Coq proof in <proof> tags using your analysis. Do not

- repeat the context and do no restate the theorem.
- 275 You are in the middle of the proof of involution_injective:

```
forall f : nat -> nat,
(forall n : nat, n = f (f n)) -> forall n1 n2 : nat, f n1 = f n2 -> n1 = n2
```

276 Ready? Here is the current goal.

```
<goal>
f : nat -> nat
H : forall n : nat, n = f (f n)
n1 : nat
n2 : nat
Hf_eq : f n1 = f n2
|- n1 = n2
</goal>
```

²⁷⁷ Take a deep breath and walk me through the proof.

278 **B** Detailed results

279 B.1 Logical Foundations

For the template agent, the gray numbers indicate the proportion of proofs that are correct at the first try (no holes). We also report the average length of the generated proof (number of tactics) and the size of the smallest and the biggest proof.

	tactics		template			tac	tactics	tactics tem
	naive	beam	naive	beam		 naive	naive beam	naive beam naive
b_true_elim2	6	5	10	10	subseq_app	4	4 4	
.ower_letter_lowers	x	8	27	8	subseq_trans	X		
grade_lowered_once	x	8	9	15	reflect_iff	13		
ade_fowered_once	x	X	×	×	eqbP_practice	x		
unt_member_nonzero	x	x	x	x	merge_filter	x	x 4	× 4 4
move_does_not_increase_count	x	x	x	x	pal_app_rev	x		
volution_injective	7	x	9	7	pal_rev	7	7 4	7 4 4
ion_elim_hd	x	x	x	x	palindrome_converse	x	x x	x x x
b_id_refl	x	6	22	18	pigeonhole_principle	×		
date_eq	14	12	×	14	regex_match_correct	x		
date_neg	7	7	7	7	rev_involutive	10		
l_comm	x	12	x	x	map_rev	x		
n_S	x	x	x	41	uncurry_curry	x		
huffle3	x	x	x	x	curry_uncurry	x		
omm	х	x	x	x	cevalceval_step	x		
_leb_compat_l	x	x	x	x	leb_plus_exists	X		
plus_distr_r	x	x	x	x	In_map_iff	31		
assoc	x	12	x	x	In_app_iff	x		
shuffle3'	13	x	x	x	All_In	x		
to_nat_pres_incr	x	x	x	x	combine_odd_even_intro	x		
_bin_nat	x	x	x	x	combine_odd_even_elim_odd	x		
_nat_bin	x	x	x	х	combine_odd_even_elim_even	x		
mize_0plus_b_sound	x	x	x	х	eqb_neq	x		
o_2_ceval	x	x	x	х	eqb_list_true_iff	x		
never_stops	x	x	x	х	forallb_true_iff	x		
hiles_eqv	x	x	x	х	tr_rev_correct	× 18		
ute_app	x	x	x	x	excluded_middle_irrefutable			
pile_correct	x	x	x	х	total_relation_not_partial_function	× 15		
k_ignore	4	4	4	4	lt_trans'			
_continue	4	4	6	6	lt_trans''	11		
stops_on_break	x	4	x	х	le_S_n	× 10		
ontinue	x	x	x	x	le_not_symmetric	10		
ops_on_break	4	4	x	х	le_antisymmetric	14 9		
_break_true	4	4	x	6	le_step	-		
_deterministic	x	8	9	3	rtc_rsc_coincide	x		
le	8	8	13	11	<pre>booltree_ind_type_correct</pre>	X		
nsense	6	7	x	21	Toy_correct	x		
/	x	x	x	x	reflect_involution	×		
_plus	x	x	x	х	t_update_neq	7		
relation_is_total	x	x	x	x	t_update_permute	X		
relation_is_empty	5	5	x	5	rev_exercise1	9	, .	
	4	4	9	4	eqb_true	x		
e_Smn_le_m	5	8	5	10	plus_n_n_injective	х		
_cases	x	7	x	х	combine_split	x		
us_1	6	6	x	10	<pre>bool_fn_applied_thrice</pre>	×		
le	x	x	x	x	eqb_sym	x		
e_cases	x	14	x	x	eqb_trans	×		
e_compat_r	x	14	x	9	split_combine	x		
s_trans	6	6	x	10	existsb_existsb'	x		
nn_le_m	x	7	11	8	ev_8	7		
t	x	×	X	x	pe_implies_pi	x		
mplete	x	x	23	25	ev100	x		
rrect	x	x	23 X	23 X	andb3_exchange	x		
ue_trans	7	7	x	9	andb_true_elim2	4		
uiv_fR	×	×	x	×	andb3_exchange'	x		
eq_refl	x	x	x	x	nor_comm'	14		
cy_1 c1 1	×	×	×	X	nor_not'	11	11 9	11 9 19

Table 1: Detailed results for the Logical Foundations benchmark.

		tac	tics	temj		
		naive	beam	naive	beam	total
283	# success	34	46	(19) 33	(26) 52	113
	% success	30.1	40.7	29.2	46.0	100.0
	average proof length	9.1	8.13	14.4	15.4	
	(min, max) proof length	(4, 31)	(4, 21)	(4, 52)	(3, 69)	

284 **B.2** BB(4)

For each methods, we also report the original proof sizes (mean, min, and max) on the set of lemmas that was successfully proved.

Ffx_eq_x_inj 10 9 7 HaltSAt_swap enc_plar_inj 12 x HaltTimeUpperBound_LE_swap_InitES enc_plar_inj 16 x x Trans_rev_rev andb_shortcut_spec 3 7 7 option_Trans_rev_rev andb_shortcut_spec 3 7 7 option_Trans_rev_rev andb_shortcut_spec 3 7 7 option_Trans_rev_rev appLsk_len 8 x fras_rev_rev app_back_len 8 x fext_inv app_back_upush_back 6 x x fext_inv app_back_upush_back 6 x x fext_ev_0 sigma_equ_spec 3 3 LE_rev_0 11 12 x HaltSAt_rev_0 sigma_enc_inj 12 x x HaltSAt_rev_0 11	1		orig.	naive	
nc_pair_inj 12 x x HaltTimeUpperBound_LE_swap_InitES nc_list_inj 16 x x Trans_rev_rev ndb_shortcut_spec 3 7 TM_rev_rev et_ins_spec 3 x x Tape_rev_rev mpty_set_WF 10 24 18 ExecState_rev_rev op_back_inth_error 15 x fst_pe_rev op_back_inth_error 3 3 Steps_rev t_edo_spec 3 3 LE_rev op_back_ingust 6 x x igma_eqb_spec 3 3 LE_rev igma_enc_inj 2 x k HaltStrev sitsigm_inj 12 x HaltStrev ap_inj 9 27 C6 HaltTimeUpperBound_LE_rev_InitES it_stspec 4 13 Its step_lousedState Its step_lousedState ifing_aper_op 3 3 Steps_lousedState Its steps_lousedState orallb_St_spec 9 x HaltTimeUpperBound_LE_HaltAtES_UnusedState orallb_st_spec 9			9	x	
nc_list_inj 16 x x Trans_rev_rev ndb_shortcut_spec 3 7 option_Trans_rev_rev rb_shortcut_spec 3 9 7 Mr_rev_rev et_ins_spec 33 x x Tape_rev_rev op_back_len 8 x x fext_inv op_back_mth_error 15 x step_halt_rev op_back_mth_error 34 48 x step_rev ist_eq_nth_error 34 48 x step_halt_rev op_back_mth_error 34 48 x step_halt_rev_0 it_eqb_spec 3 3 1E_rev_0 it_eqb_spec 3 3 3 1E_rev_0 istSigma_inj 12 x 48 HaltTimeUpperBound_LE_rev_InitES ifm_edp_spec 4 12 26 isUnusedState_spec istime_operound_LE_rev_initES ifm_list_gpec			10	x	
nc_list_inj 16 x x Trans_rev_rev ndb_shortcut_spec 3 7 7 option_Trans_rev_rev t_ins_spec 3 9 7 TM_rev_rev et_ins_spec 3 x x Tape_rev_rev pp_back_len 8 x x fext_inv op_back_len 8 x x fext_inv op_back_unth_error 15 x x step_rev t_enc_inj 2 37 37 LE_rev_0 t_enc_inj 2 37 37 LE_rev_0 igma_eqb_spec 3 x x InitES_rev igma_encinj 2 x x HaltsAt_rev_0 istSigma_inj 9 27 26 HaltimeUpperBound_LE_rev_InitES ir_eqb_spec 3 3 3 13 Trans_swapid t_ists_pec 4 12 26 isUnusedState ir_list_spec 4 12 26 isUnusedState orallb_Sigma_spec 9 x 14 HaltimeUpperBound_LE_HaltAtES_UnusedState orallb_Sigma_spec 9 x 3 TM0_LE orallb_Sigma_spec 9 x 14 HaltimeUpperBound_LE_HaltAtES_MergeUnusedState orallb_Sigma_spec 9 x 3 TM0_LE teps_unique 11 22 x 48 LatitimeUpperBound_LE_rev_InitES ir_eqb_spec 4 12 26 isUnusedState orallb_Sigma_spec 9 x 14 HaltIimeUpperBound_LE_HaltAtES_UnusedState orallb_Sigma_spec 9 x 3 TM0_LE orallb_Sigma_spec 9 x 14 HaltIimeUpperBound_LE_HaltAtES_UnusedState orallb_Sigma_spec 9 x 3 TM0_LE teps_unique 11 22 x 48 UnusedState_te orallb_Sigma_spec 9 x 14 HaltIimeUpperBound_LE_HaltAtES_MergeUnusedState orallb_Sigma_spec 9 x 8 NuseState_te orallb_Sigma_spec 9 x 8 NuseState_te orallb_Sigma_spec 9 x 8 NuseState_dec teps_unique 11 22 x KaltItimeUpperBound_LE_HaltAtES_MergeUnusedState indial_tiff 27 x x St_suc_eq E_step 10 39 18 St_suc_neq E_steps 10 34 X HaltItimeUpperBound_LE_HaltAtES_UnusedState_te intimeUpperBound_LE_HaltAtES_UnusedState_te_trupd LitsAtUnique 16 x x St_suc_le onHaltItrans_0 altItimeUpperBound_LE_HaltAtES_UnusedState_te_trupd L_HaltStatES_1 11 x X UnusedState_trupd L_HaltStatES_2 14 x x isHaltTrans_0 altTimeUpperBound_LE_HaltAtES_UnusedState_te_trupd L_HaltStatES_1 11 x X UnusedState_trupd L_HaltStatES_1 11 X X UnusedState_trupd L_HaltStatES_2 14 X X CountHaltTrans_0 ANNHALT			5	х	
rb_shortcut_spec 3 9 7 TM_rev_rev t_ins_spec 33 x x Tape_rev_rev op_back_len 8 x x fext_inv op_back_inth_error 15 x x step_rev op_back_inth_error 34 48 x step_rev op_back_inth_error 34 48 x step_rev ist_eqnth_error 34 48 x step_rev igma_ent_inth_error 3 3 1E_rev_0 igma_ent_inth_error 3 3 1E_rev_0 istigma_int 12 x k HaltstimeUpperBound_LE_rev_IntES ir_eqt_spec 3 3 1Trans_swap_id 1 t_istspec 4 13 3 step_UnusedState_spec 1 <			7	15	
rb_shortcut_spec 3 9 7 Tw_rev_rev st_ins_spec 33 x x Tape_rev_rev pty_set_WF 10 24 18 ExecState_rev_rev pp_back_len 8 x fext_inv pp_pback_rev pp_back_unth_error 15 x step_rev ist_eqnth_error 34 48 x step_rev pp_back_unth_error 34 48 x step_rev igma_eqh_pback_unth_error 34 48 x step_rev igma_eqh_spec 3 3 LE_rev_0 trans_step_untot igma_eqh_spec 3 3 LE_rev_0 trans_step_untot ististigm_inj 12 x 48 Haltstat_rev_0 ististististic_ing 7 7 8 HaltTimeUpperBound_LE_rev_InitES ir_eqh_spec 3 3 Istep_sunsedState_spec talt_instupperBound_LE_rev_InitES ir_eqh_spec 9 x 14 HaltTimeUpperBound_LE_HaltAtES_UnusedState <td></td> <td></td> <td>8</td> <td>10</td> <td></td>			8	10	
spec 33 x x Tape_rev_rev mpty_set_WF 10 24 18 ExecState_rev_rev mpty_set_WF 10 24 18 ExecState_rev_rev pp_back_len 8 x ffext_inv pp_back_len 6 x step_halt_rev pp_backmth_error 34 48 x step_halt_rev pp_backmth_error 34 48 x step_halt_rev pp_backmth_error 34 48 x step_halt_rev str_eq0_spec 3 3 LE_rev_0 igma_enc_inj 2 x K HaltStr_rev_0 ststigma_inj 9 27 26 HaltTimeUpperBound_LE_rev_InitES ist_reqb_spec 3 3 13 Trans_swap.id itspec 4 12 26 isUnusedState_spec igma_list_spec 9 x 14 HaltTimeUpperBound_LE_HaltsAtES_UnusedState orallb_sigma_spec 9 x 18 UnusedState_dec teps_unique 11 22 <t< td=""><td></td><td></td><td>7</td><td>10</td><td></td></t<>			7	10	
pty_set_WF 10 24 18 ExecState_rev_rev p_back_len 8 x fext_inv p_back_len 8 x fext_inv p_back_len 34 48 x step_halt_rev p_back'_pushback 6 x step_rev _enc_inj 2 37 37 LE_rev_0 _eqb_spec 3 x InitES_rev gma_enc_inj 2 x HaltsimeUpperBound_LE_rev stSigma_inj 12 x 48 HaltsimeUpperBound_LE_rev stSigma_inj 12 x 48 HaltsimeUpperBound_LE_rev stI_enc_inj 7 7 8 HaltTimeUpperBound_LE_rev st_spec 3 13 Trans_swap_id _list_spec 4 13 Steps_UnusedState railb_Sigma_spec 9 x 14 HaltTimeUpperBound_LE_HaltsAtES_UnusedState railb_Sigma_spec 9 x 18 UnusedState_TMO eps_unique 11			7	х	
p.pback_len8xfext_invp.pback_nth_error15xstep_revsit_eqnth_error3448xste_qnth_error3448xste_qnth_error3448xste_qnth_error3448xste_qnth_error3437LE_rev_0eqb_spec33LE_rev_0gma_edp_spec33Sgma_enc.inj2xXstSigma_inj12x48HaltsineUpperBound_LE_revIstSigma_inj92726HaltTimeUpperBound_LE_rev_InitESstr_eqdp.spec3313Trans_swap_id:_list_spec41226isUnusedState_specgma_list_spec9x13step_UnusedStateorallb_Sigma_spec9x16UnusedState_deceps_trans9x16UnsedState_deceps_NoHalt22xxStepsitst.grep1034xHaltTimeUpperBound_LE_HaltAtES_UnusedStateorallb_Sigma_spec9x16UnusedState_deceps_Unique16xx <tsuc_eq< td="">itstat_unique16xx<tsuc_eq< td="">itstat_unique16xx<tsuc_eq< td="">itstat_unique16itstate_unique<</tsuc_eq<></tsuc_eq<></tsuc_eq<>			7	x	
			3	5	
ist_eqnth_error 34 48 x step_halt_rev gp_back'push_back 6 x x Steps_rev _enc_inj 2 37 37 LE_rev.0 :_eqb_spec 3 3 LE_rev.0 :gma_eqb_spec 3 x x InitES_rev istSigma_inj 12 x HaltsAt_rev_0 istSigma_inj 9 27 26 HaltTimeUpperBound_LE_rev_InitES istI_enc_inj 7 7 8 HaltTimeUpperBound_LE_rev_InitES ir_eqb_spec 3 3 13 Trans_swap_id list_spec 4 12 26 isUnusedState ir_list_spec 4 13 15 step_lunusedState orallb_Sim_spec 9 x 33 TM0_LE orallb_ir_spec 9 x St_suc_eq orallb_ir_spec 9 x St_suc_eq orallb_ir_spec 9 x St_suc_eq orallb_ir_spec 9 x St_suc_eq orallb_ir_rev 10 39			44	x	
pp_back'push_back 6 x x Steps_rev _enc_inj 2 37 37 LE_rev_0 _eqb_spec 3 3 LE_rev_0 gma_eqb_spec 3 X InitES_rev gma_enc_inj 2 x HaltSAt_rev_0 stSigma_inj 12 x HaltAtAt_rev p.inj 9 27 26 HaltTimeUpperBound_LE_rev stT_enc_inj 7 7 8 HaltTimeUpperBound_LE_rev stt_epsec 3 3 Trans_swap_id _list_spec 4 12 26 isUnusedState r_allb_St_spec 4 13 Steps_UnusedState 13 r_allb_st_spec 9 x 13 TMo_LE rallb_sigma_spec 9 x 14 HaltTimeUpperBound_LE_HaltAtES_UnusedState rallb_ir_spec 9 x 18 UnusedState_dec reps_nonque 11 22 x St_sto_ant reps_NonHalt			11	x	
eqc_inj 2 37 37 LE_rev_0 eqc_spec 3 3 LE_rev gma_eqb_spec 3 x InitES_rev gma_enc_inj 2 x HaltsAt_rev_0 stSigm_inj 12 x HaltsAt_rev_0 stSigm_inj 2 x HaltTimeUpperBound_LE_rev stSigm_inj 7 7 8 HaltTimeUpperBound_LE_rev_InitES r.redp_spec 3 3 13 Trans_swap_id _list_spec 4 12 26 isUnusedState_spec gma_list_spec 4 13 Steps_UnusedState rrllb_Sigma_spec 9 x 14 HaltTimeUpperBound_LE_HaltsAtES_UnusedState orallb_Sigma_spec 9 x 18 UnusedState_dec 10 veps_unique 11 22 x St_suc_le 11 ueps_wohnHalt 22 x St_suc_le 11 ueps_wohnHalt 22 x St_suc_le 11 <tr< td=""><td></td><td></td><td>27</td><td>x</td><td></td></tr<>			27	x	
eqb_spcc 3 3 3 LE_rev gma_edp_spcc 3 x x InitES_rev gma_enc_inj 2 x x HaltSAt_rev_0 stSigma_inj 12 x 48 HaltSAt_rev_0 p_inj 9 27 26 HaltTimeUpperBound_LE_rev stT_enc_inj 7 7 8 HaltTimeUpperBound_LE_rev gma_list_spec 3 13 Trans_swap_id ists_spec 4 12 26 isUnusedState rallb_St_spec 4 13 Steps_UnusedState rallb_st_spec 9 x 14 HaltTimeUpperBound_LE_HaltAtES_UnusedState rallb_isgma_spec 9 x 18 UnusedState_dcc reps_unique 11 22 x St_suc_at eps_unique 11 22 x St_suc_at eps_unique 16 x St_suc_at istAt_unique 16 x St_suc_at istep<			7	x	
gma_eqb_spec3xxInitES_revgma_enc_inj2xxHaltAt_rev_0stSigma_inj12x48HaltAt_revup_inj92726HaltTimeUpperBound_LE_rev_InitESist_enc_inj778HaltTimeUpperBound_LE_rev_InitESir_eqb_spec3313Trans_swap_idi_list_spec41226isUnusedStateir_list_spec413Steps_UnusedStateir_list_spec9x14ir_list_spec9x33trallb_Sigma_spec9x33trallb_ir_spec916UnusedState_deceps_trans9x18UnusedState_deceteps_trans9x18UnusedState_decitststunique16xSt_suc_leintst_unique16xSt_suc_leintst_unique16xSt_suc_leintimeUpperBound_LE_NonHalt7xUnusedState_ptr_NonHalts8xHaltStTES_TransintimeUpperBound_LE_NonHalt7xCountHaltTrans_0intTimeUpperBound_LE_Halt15xCountHaltTrans_0intTimeUpperBound_LE_Halt15xCountHaltTrans_0intTimeUpperBound_LE_Halt15xCountHaltTrans_0intTimeUpperBound_LE_NonHalt7xNuses_swap7x8Trans_swap_inditTimeUpperBound_LE_NonHalt15 <td></td> <td></td> <td>9</td> <td>x</td> <td></td>			9	x	
Igma_enc_inj2xXHaltsAt_rev_0istSigm_inj12x48HaltsAt_revsp_inj92726HaltTimeUpperBound_LE_rev_InitESistT_enc_inj778HaltTimeUpperBound_LE_rev_InitESir_edp_spec3313Trans_swap.idistspec41226isUnusedState_specigma_list_spec41313step_UnusedStateorallb_Sigma_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Sigma_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_sigma_spec9x18UnusedState_decteps_trans9x18UnusedState_decitstyperspecs103918St_suc_leonHalt_iff27xSt_suc_leitsthunique16xXitsthuff11xUnusedState_unditstheUpperBound_LE_NonHalt7xSt_suc_leitstheUpperBound_LE_NonHalt7xSt_suc_leitstheUpperBound_LE_NonHalt7xUnusedState_ptr_unditaskasp_swap12xCountHaltTrans_0ittimeUpperBound_LE_Halt15xCountHaltTrans_0itstasp=swap71011St_leb_specitastasp=swap78Trans_list_specitastasp=swap78Trans_list_specitastasp=swap78Trans_list_spec <t< td=""><td></td><td></td><td>3</td><td>13</td><td></td></t<>			3	13	
istSigm_inj12x48HaltsAt_revap.inj92726HaltTimeUpperBound_LE_revistT_enc_inj778HaltTimeUpperBound_LE_rev_InitESir_eqb_spec3313Trans_swap_idtlist_spec41226isUnusedState_specigm_list_spec4133Step_UnusedStateir_list_spec413Step_UnusedStateorallb_St_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Simm_spec9x33TM0_LEorallb_Simm_spec9x18UnusedState_decteps_trans9x18UnusedState_decteps_unique1122xSt_suc_leonHalt22xxSt_suc_le_steps_NoHalt22xXSt_suc_le_steps103918St_suc_leq_steps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptr_NoHalts8xxHaltStE_g1_litimeUpperBound_LE_NonHalt7xKusedState_lor12xCountHaltTrans_0altTimeUpperBound_LE_Halt15xCountHaltTrans_upd_swap_swap71011St_leb_spec_swap_swap766TM_upd'_spec_swap_swap8xxnat_eqb_spec_tien_berned1127xTNF_Node_expand_spec			15	17	
ap_ini92726HaltTimeUpperBound_LE_revistT_enc_inj778HaltTimeUpperBound_LE_rev_InitESistT_enc_inj778HaltTimeUpperBound_LE_rev_InitESir_eqb_spec3313Trans_swap_idt_list_spec41226isUnusedState_specigma_list_spec413Steps_UnusedStateorallb_Sigma_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Dir_spec9xorallb_Dir_spec9xteps_trans9xteps_unique1122xHaltTimeUpperBound_LE_HaltAtES_MergeUnusedStateorallb_liff27xstst_unique16xstst_unique16xststst_unique103918st_steps103918st_steps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptrE_NonHalts8xtimeUpperBound_LE_NonHalt7xUnusedState_undLHaltsAtES_111xundeState_ptr_updLawaper7xtrans_wap7xCountHaltTrans_0altTimeUpperBound_LE_Halt15xCountHaltTrans_0_NonHaltrans_swap_swap7xxSt_specy1011st_spec10y14x <td></td> <td></td> <td>9</td> <td>x</td> <td></td>			9	x	
stT_enc_inj778HaltTimeUpperBound_LE_rev_InitESir_edp.spec3313Trans_swap.id_list_spec41226isUnusedState_specigma_list_spec41313step_UnusedStateorallb_Sigma_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Sigma_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Sigma_spec9x33TM0_LEorallb_Sigma_spec9x18UnusedState_decceps_trans9x8UnusedState_decceps_unique1122xSt_suc_leuntationuc16xxSt_suc_leonHalt_iff27xxSt_suc_leSteps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptriNnHalts8xxHaltsAtES_TransiltTimeUpperBound_LE_NonHalt7xWusedState_ptri_AltsAtES_111xWusedState_ptri_HaltsAtES_214xisHaltTrans_0iltTimeUpperBound_LE_Halt15xCountHaltTrans_Updi_swap_swap78Trans_list_speci_swap_swap767M_simplify_speci_swap_swap766TM_upd'_speci_swap_swap766TM_upd'_speci_swap_swap766TM_upd'_speci_swap_swap76			10	x	
ir_eqb_spec3313Trans_swap_idi_lits_spec41226isUnusedState_specigma_list_spec41313step_UnusedStateir_list_spec41313Steps_UnusedStateir_list_spec9x13TM0_LEorallb_Sigma_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Dir_spec916UnusedState_Im0ceps_unique1122xHaltTimeUpperBound_LE_HaltAtES_MergeUnusedStateceps_unique1122xHaltTimeUpperBound_LE_HaltAtES_MergeUnusedStateceps_unique16xxSt_suc_leonHalt22xXSt_co_nateps_wonHalt22xxSt_suc_leistep103918St_suc_eqistep1034xHaltAtteS_UnusedState_ptr_NoHAlts8xxHaltAtES_TransitTimeUpperBound_LE_Halt7xKiHaltsAtES_111xUnusedState_updiHaltsAtES_214xisHaltTrans_0itTimeUpperBound_LE_Halt15xCountHaltTrans_upd_swap_swap71011st_leb_speci_swap_swap766TM_upd'_speciceState_swap_swap766TM_upd'_specicep_spath8xxint_eqb_specicep_swap8xxint_eqb_spec			5	x	
:_list_spec 4 12 26 isUnusedState_spec gma_list_spec 4 13 13 step_UnusedState :r_list_spec 4 x 13 Steps_UnusedState :r_list_spec 9 x 14 HaltTimeUpperBound_LE_HaltsAtES_UnusedState :rallb_Sigma_spec 9 x 33 TM0_LE orallb_Dir_spec 9 17 16 UnusedState_TM0 :eps_trans 9 x 18 UnusedState_dec :eps_trans 9 x 18 UnusedState_dec :eps_unique 11 22 x HaltTimeUpperBound_LE_HaltAtES_MergeUnusedState :eps_unique 16 x x St_suc_le :steps 10 39 18 St_suc_eq :steps 10 39 4 x HaltTimeUpperBound_LE_HaltAtES_UnusedState_ptr :NonHalts 8 x x HaltS_Trans :ItTimeUpperBound_LE_NonHalt 7 x UnusedState_und :HaltsAtES_1 11 x x UnusedState_und :HaltsAtES_1 11 x x UnusedState_und :HaltsAtES_1 11 x x CountHaltTrans_0 :ItTimeUpperBound_LE_Halt 15 x CountHaltTrans_0, ItTimeUpperBound_LE_Halt 15 x CountHaltTrans_0, ItTimeUpperBound_LE_Halt 15 x CountHaltTrans_0, ItTimeUpperBound_LE_Halt 15 x CountHaltTrans_0, ItTimeUpperBound_LE_Halt 15 x X CountHaltTrans_0, ItTimeUpperBound_LE_Halt 22 X X CountHaltTrans_0, ItTimeUpperBound_LE_Halt 25 X X CountHaltTrans_0, ItTimeUpperBound_LE_Halt 25 X X CountHaltTrans_0, ItTimeUpperBound_LE_Halt 25 X X CountHaltTrans_0, ItTimeUpperBound_LE_Halt 25 X X X X X X X X X X X X X X X X X X			10	x	
gma_list_spec41313step_UnusedStater_list_spec4x13Steps_UnusedStateorallb_Sigma_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Dir_spec9x33TM0_LEorallb_sigma_spec9x33TM0_LEorallb_loir_spec91716UnusedState_decceps_trans9x18UnusedState_decceps_unique1122xSt_tc_nat_injlitst_unique16xxSt_suc_leuntational16xxSt_suc_leonHalt_iff27xxSt_suc_leSteps1034xHaltTimeUppeRound_LE_HaltAtES_UnusedState_ptrNonHalts8xxHaltsAtES_TransiltTimeUpperBound_LE_NonHalt7xKunusedState_und11xunusedState_undiHaltsAtES_111xunusedState_undi_HaltsAtES_214xisHaltTrans_0iltTimeUpperBound_LE_Halt15xCountHaltTrans_0intTimeUpperBound_LE_Halt15xCountHaltTrans_0ittincupseqswap7x8Trans_list_speciswap_swap767M_simplify_specieeState_swap_swap766TM_uof_specieeState_swap_swap766TM_uof_specieep_swap1027xTNF_Node_expand_spec <td></td> <td></td> <td>58</td> <td>x</td> <td></td>			58	x	
Initial spec4x13Steps_UnusedStateorallb_St_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStateorallb_Sigma_spec9x33TM0_LEorallb_Sigma_spec9x16UnusedState_decups_trans9x18UnusedState_decups_trans9xSt_suc_decups_trans9xSt_suc_decups_trans9xSt_suc_decups_trans1022xSt_suc_decups_trans16xxSt_suc_decups_trans103918St_suc_eqistep1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptr_NoHalts8xHaltsAtES_Trans_NoHalts8xHaltsAtES_trans_HitimeUpperBound_LE_NonHalt7xUnusedState_ptr_upd_HaltsAtES_111xUnusedState_ptd_HaltsAtES_214xisHaltTrans_0ultTimeUpperBound_LE_Halt15xCountHaltTrans_upd_swap_swap71011st_leb_spec_swap_swap766TM_upd'_spec(ecState_swap_swap766TM_upd'_specups_balt_swap1027xTNF_Node_expand_specups_balt_swap27xTNF_Node_NonHalt			11	14	
rallb_St_spec9x14HaltTimeUpperBound_LE_HaltsAtES_UnusedStaterallb_Sigma_spec9x33TM0_LErallb_Dir_spec91716UnusedState_TM0eps_trans9x18UnusedState_deceps_trans9x18UnusedState_deceps_noHalt22xXSt_to_nat_injltsAt_unique16xSt_suc_lenHalt_iff27xXSt_suc_eq_step103918St_suc_eq_steps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptr_NonHalts8xHaltStES_TransltTimeUpperBound_LE_NonHalt7xUnusedState_upd_HaltsAtES_111xVunusedState_ptr_upd_HaltsAtES_214xisHaltTrans_0ltTimeUpperBound_LE_Halt15xCountHaltTrans_upd_swap_swap71011St_leb_spec_swap_swap766TM_upd'_spececState_swap_swap766TM_upd'_specep_halt_swap18xxnat_eqb_specep_s_swap27xXTTF_Node_NonHalt			15	14 X	
rallb_Sigma_spec9x33TM0_LErallb_Dir_spec91716UnusedState_TM0eps_trans9x18UnusedState_deceps_unique1122xHaltTimeUpperBound_LE_HaltAtES_MergeUnusedStateeps_unique1122xSt_to_nat_injltsAt_unique16xxSt_suc_lenHalt_iff27xxSt_suc_le_step103918St_suc_neq_steps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptr_NonHalts8xxHaltsAtES_Trans1tTimeUpperBound_LE_NonHalt7xWusedState_upd_HaltsAtES_111xWusedState_ptr_upd_HaltsAtES_214xisHaltTrans_01tTimeUpperBound_LE_Halt15xCountHaltTrans_0_mas_swap_swap7xNtion_rrans_swap_swap7xRrans_swap_swap76TM_simplify_spececState_swap_swap766ep_savap8xnat_eqb_specep_halt_swap1027xTMF_Node_NonHalt			68	x	
rallb_Dir_spec91716UnusedState_TM0eps_trans9x18UnusedState_deceps_unique1122xHaltTimeUpperBound_LE_HaltAtES_MergeUnusedStateeps_NonHalt22xxSt_to_nat_injltsAt_unique16xxSt_suc_lenHalt_iff27xxSt_suc_neq_step103918St_suc_neq_steps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptr_NonHalts8xxHaltStES_TransItTimeUpperBound_LE_NonHalt7xx_HaltsAtES_111xx_HaltsAtES_214xx_swap_swap7x& CountHaltTrans_0ito_mrans_swap_swap71011_ste_essap766_swap_swap766_swap_swap18xx_swap_swap18xx_swap_swap766_swap_swap18xx_swap_swap27xX_swap_swap766_swap_swap18x_swap_swap77xxxXxXxXxXxXxXxXxXxXxXxX <td></td> <td></td> <td>7</td> <td>5</td> <td></td>			7	5	
eps_trans9x18UnusedState_deceps_unique1122xHaltTimeUpperBound_LE_HaltAtES_MergeUnusedStateeps_NonHalt22xXSt_to_nat_injtisAt_unique16xX St_suc_lehHalt_iff27xXSt_suc_eq_step103918St_suc_neq_Steps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptrNonHalts8xxHaltAtES_TransltTimeUpperBound_LE_NonHalt7xUnusedState_updHaltsAtES_111xxUnusedState_updHaltsAtES_214xisHaltTrans_0ltTimeUpperBound_LE_Halt15xCountHaltTrans_upd_swap_swap7x8Trans_list_spection_Trans_swap_swap71011St_leb_spec_swap_swap8xxnat_eqb_specepsape18xxnat_eqb_specep_halt_swap1027xTWF_Node_expand_speceps_swap27xXTWF_Node_NoHAlt			10	22	
Personique1122xHaltTimeUpperBound_LE_HaltAtES_MergeUnusedStatepps_NonHalt22xxSt_to_nat_injltsAt_unique16xxSt_suc_lenHalt_iff27xxSt_suc_lestep103918St_suc_neq_step1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptrNonHalts8xxHaltSAtES_TransltTimeUpperBound_LE_NonHalt7xunusedState_updHaltsAtES_111xxUnusedState_ptrHaltsAtES_214xxisHaltTrans_0ltTimeUpperBound_LE_Halt15xCountHaltTrans_upd_swap_swap12xxCountHaltTrans_oNalt_swap_swap71011St_leb_spec_swap_swap766TM_upify_specceState_swap_swap766TM_upif_specceState_swap_swap18xxnat_eqb_specep_halt_swap1027xTMF_Node_NonHalt			4	X	
eps_NonHalt22xSt_to_nat_injltsAt_unique16xxSt_suc_lenHalt_iff27xxSt_suc_leq_step103918St_suc_neq_step1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptrNonHalts8xHaltAtES_TransITTimeUpperBound_LE_NonHalt7xVusedState_upd_HaltsAtES_111xVusedState_ptr_upd_HaltsAtES_214xisHaltTrans_0ltTimeUpperBound_LE_Halt15xCountHaltTrans_upd_swap_swap7x8Trans_iswap_swap71011stlesswap_swap8x9TM_simplify_spececState_swap_swap7ecState_swap_swap18xnat_eqb_specep_halt_swap1027xTNF_Node_expand_specep_s_swap27xxTNF_Node_NorHalt		at 0	31	x	
hltsAt_unique 16 x x St_suc_le nhHalt_iff 27 x x St_suc_eq _step 10 39 18 St_suc_eq _step 10 34 x HaltTimeUpperBound_LE_HaltAtES_UnusedState_ptr _NonHalts 8 x x HaltAtES_Trans ltTimeUpperBound_LE_NonHalt 7 x unusedState_ptr_upd _HaltAtES_1 11 x unusedState_ptr_upd _HaltAtES_2 14 x isHaltTrans_0 ltTimeUpperBound_LE_Halt 15 x CountHaltTrans_upd _swap_swap 12 x CountHaltTrans_oNonHalt cas_swap_swap 7 10 11 St_leb_spec uswap_swap 8 x 9 TM_simplify_spec cecState_swap_swap 10 27 x TNF_Node_expand_spec uep_halt_swap 27 x TNF_Node_NonHalt		ale	4	4	
onHalt_iff 27 x St_suc_eq _step 10 39 18 St_suc_nq _steps 10 34 x HaltimeUpperBound_LE_HaltAtES_UnusedState_ptr E_NonHalts 8 x HaltsAtES_Trans ltTimeUpperBound_LE_NonHalt 7 x UnusedState_upd LHaltsAtES_1 11 x UnusedState_upd LHaltsAtES_2 14 x isHaltTrans_0 ltTimeUpperBound_LE_Halt 15 x CountHaltTrans_upd :_swap_swap 12 x CountHaltTrans_0_NonHalt itrineupperBound_LE_Halt 15 x CountHaltTrans_0_NonHalt :_swap_swap 7 x 8 Trans_list_spec :_swap_swap 7 10 11 St_leb_spec (_swap_swap 7 6 6 TM_upi_spec :eeCstate_swap_swap 7 6 6 TM_upi_spec :eep_swap 10 27 x TNF_Node_expand_spec :eps_swap 27 x TNF_Node_NonHalt			4	×	
istep103918St_suc_neqisteps1034xHaltTimeUpperBound_LE_HaltAtES_UnusedState_ptristeps1034xHaltAtES_TransintTimeUpperBound_LE_NonHalt7xxitTimeUpperBound_LE_NonHalt7xUnusedState_updi_HaltsAtES_111xUnusedState_ptr_updi_HaltsAtES_214xisHaltTrans_0itTimeUpperBound_LE_Halt15xCountHaltTrans_updisswap_swap12xcountHaltTrans_peciton_Trans_swap_swap7x8iton_Trans_swap_swap71011its_leb_spec1027xitecState_swap_swap18xnat_edp_speciep_swap27xTMF_Node_expand_speciep_sap27xTNF_Node_NonHalt			5	x	
			3	7	
NonHalts 8 x HaltsAtES_Trans lhTimeUpperBound_LE_NonHalt 7 x UnusedState_upd HaltsAtES_1 11 x UnusedState_upd LHaltsAtES_2 14 x isHaltTrans_0 lhTimeUpperBound_LE_Halt 15 x CountHaltTrans_0MonHalt :_swap_swap 12 x CountHaltTrans_0_MonHalt ans_swap_swap 7 x B titon_Trans_swap_swap 7 10 11 titon_Trans_swap_swap 7 6 6 usequestate_swap_swap 7 6 6 tecState_swap_swap 10 27 x nat_eqb_spec uep_halt_swap 27 x TMF_Node_NonHalt		tr	21	×	
iltTimeUpperBound_LE_NonHalt 7 x x UnusedState_upd :HaltsAtES_1 11 x WnusedState_upd :HaltsAtES_2 14 x isHaltTrans_0 iltTimeUpperBound_LE_Halt 15 x CountHaltTrans_upd ::swap_swap 12 x CountHaltTrans_0_NonHalt :ans_swap_swap 7 x 8 Trans_list_spec tion_Trans_swap_swap 7 10 11 St_leb_spec .swap_swap 8 9 TM_simplify_spec tecState_swap_swap 7 6 6 TM_upd'_spec :ep_smap 18 x nat_eqb_spec :ep_sap 27 x TNF_Node_expand_spec			27	x	
E_HaltsAtES_1 11 x VuusedState_ptr_upd _HaltsAtES_2 14 x isHaltTrans_0 olltTimeUpperBound_LE_Halt 15 x CountHaltTrans_upd _swap_swap 12 x CountHaltTrans_e_NonHalt rans_swap_swap 7 x 8 Trans_list_spec otion_Trans_swap_swap 7 10 11 St_leb_spec _swap_swap 8 x 9 TM_simplify_spec cecState_swap_swap 7 6 6 TM_upd'_spec rep_swap 18 x nat_edp_spec exphalt_swap 10 27 x TNF_Node_expand_spec			68	x	
_HaltsAtES_2 14 x isHaltTrans_0 ltTimeUpperBound_LE_Halt 15 x CountHaltTrans_upd _swap_swap 12 x x CountHaltTrans_0_NonHalt ans_swap_swap 7 x 8 Trans_list_spec tion_Trans_swap_swap 7 10 11 St_leb_spec Lswap_swap 8 x 9 TM_simplify_spec cecState_swap_swap 7 6 6 TM_upd'_spec ieep_halt_swap 10 27 x TNF_Node_expand_spec ieps_swap 27 x x TNF_Node_NonHalt			97	x	
IltimeUppeFBound_LE_Halt 15 x CountHaltTrans_upd swap_swap 12 x CountHaltTrans_0_NonHalt rans_swap_swap 7 x 8 tion_Trans_swap_swap 7 10 11 State_swap_swap 8 x 9 t_swap_swap 8 x 9 t_swap_swap 7 6 6 t_swap_swap 18 x nat_eqb_spec tep_swap 10 27 x tep_st_swap 27 x TNF_Node_NonHalt			3	17	
12 x x CountHaltTrans_0_NonHalt rans_swap_swap 7 x 8 Trans_list_spec ption_Trans_swap_swap 7 10 11 St_leb_spec _swap_swap 8 x 9 TM_simplify_spec _secState_swap_swap 7 6 6 TM_upd'_spec _eep_swap 18 x nat_eqb_spec _eep_halt_swap 10 27 x TNF_Node_expand_spec _eps_swap 27 x x TNF_Node_NonHalt			7	17 X	
rans_swap_swap 7 x 8 Trans_list_spec tion_Trans_swap_swap 7 10 11 St_leb_spec 1_swap_swap 8 x 9 TM_simplify_spec tecState_swap_swap 7 6 6 TM_upd'_spec tep_swap 18 x nat_eqb_spec tep_swap 10 27 x TNF_Node_expand_spec teps_swap 27 x TNF_Node_NonHalt			21	x	
httom_Trans_swap_swap 7 10 11 St_leb_spec L_swap_swap 8 x 9 TM_simplify_spec tecState_swap_swap 7 6 6 TM_upd'_spec tep_swap 18 x nat_eqb_spec tep_swap 10 27 x TMF_Node_expand_spec tep_sapp 27 x TNF_Node_NorHalt			6	x	
M_swap_swap 8 x 9 TM_simplify_spec tcecState_swap_swap 7 6 6 TM_upd'spec tcep_swap 18 x nat_eqb_spec tcep_halt_swap 10 27 x TNF_Node_expand_spec tcep_sawap 27 x x TNF_Node_NorHalt			13	x	
ecstate_swap_swap 7 6 6 TM_upd'_spec ep_swap 18 x nat_eqb_spec ep_halt_swap 10 27 x TNF_Node_expand_spec eps_swap 27 x x TNF_Node_NonHalt			6	7	
ep_swap 18 x nat_eqb_spec ep_halt_swap 10 27 x TNF_Node_expand_spec eps_swap 27 x TNF_Node_NonHalt			5	9	
ep_halt_swap 10 27 x TNF_Node_expand_spec eps_swap 27 x x TNF_Node_NonHalt			3	9 X	
eps_swap 27 x x TNF_Node_NonHalt			5 64		
			64 6	x	
			0 7	x	
			74	x	
			30	x	
nitES_swap 8 x 12 SearchQueue_upd_bfs_spec altsAt_swap_0 15 19 x SearchQueue_reset_spec			30 13	× 38	

Table 2: Detailed	results for	the $BB(4$) benchmark.
-------------------	-------------	------------	--------------

		tem		
		naive	beam	total
	# success	(19) 35	(40) 50	113
287	% success	35.0	40.0	100.0
	average proof length	16.2	14.4	
	original average proof length	7.9	7.1	
	(min, max) proof length	(3, 48)	(3, 48)	
	original (min, max) proof length	(2, 34)	(2, 27)	