Introduction

A blockchain enables peer-to-peer digital transactions without the need for a trusted intermediary. This is only possible because of its **consensus protocol**, which allows nodes within the system to agree on the state of the blockchain, even in the presence of adversaries.
For this reason, it is paramount that consensus is designed and implemented correctly to prevent the system from reaching unwanted states that can be exploited by adversaries.
The most famous example of this is Bitcoin's **Proof of Work (PoW)** consensus protocol and its susceptibility to a **51% attack**, where adversaries control the majority of the compute power in the system, which gives them the potential to double spend.
Infamous examples of such attacks include Ethereum Classic [**?**], Bitcoin Gold [**?**], and Vertcoin [**?**], totalling losses of over $30 million.
Other key components of modern blockchain systems are **bridging protocols** for cross-chain data transfer and **smart contracts** for automated agreement execution. These components are also not without their exploits, with infamous examples such as the Poly Network [**?**],

Background

A blockchain is a decentralised ledger that allows two parties to carry out transactions without the need of a trusted intermediary, eliminating the need for trust. This is only possible through a blockchain's consensus protocol, which allows all parties to agree on the current state of the blockchain and the transactions recorded on it. The most popular consensus protocol is Proof of Work (PoW) used by Bitcoin's blockchain, which has around 1.2 billion recorded transactions [**?**] with Bitcoin's market capitalisation sitting around $1.8 trillion [**?**]. The core idea of PoW is that the longest blockchain is

Isabelle is a proof assistant written in Scala and ML that uses Higher-Order Logic (HOL). It is used to write and verify formal proofs with high assurance due to the mechanisation of these proofs [**?**]. Isabelle's Isar proof language allows these proofs to be more readable than the traditional approach to theorem proving by repeatedly applying tactics. Isabelle also makes use of automation tools like Sledgehammer, which uses external automated theorem provers (ATPs) to help you complete proofs. Outside of the proof assistant itself, the Scala library Scala-Isabelle provides the functionality to interact

Related Work

Isabelle Verifications (20+ years):

- seL4 Microkernel [?]
- CakeML compiler [?]
- Protocols & programs [?, ?, ?]

Isabelle Mathematics (AFP):

- Gödel's incompleteness [?]
- Jordan curve theorem [?]
- Ramsey's theorem [?]

Recent Blockchain:

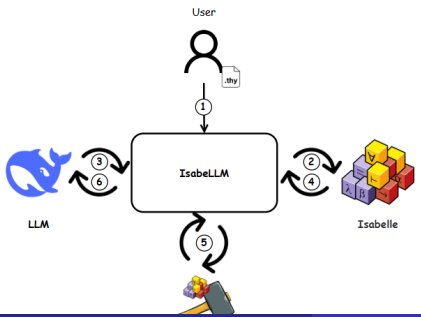
- Ethereum VM [?]
- Solidity framework [?]

Other Theorem Provers (Blockchain):

- Agda [?, ?, ?]
- Coq [?, ?, ?]
- Lean [?, ?]

DeFi & Tools:

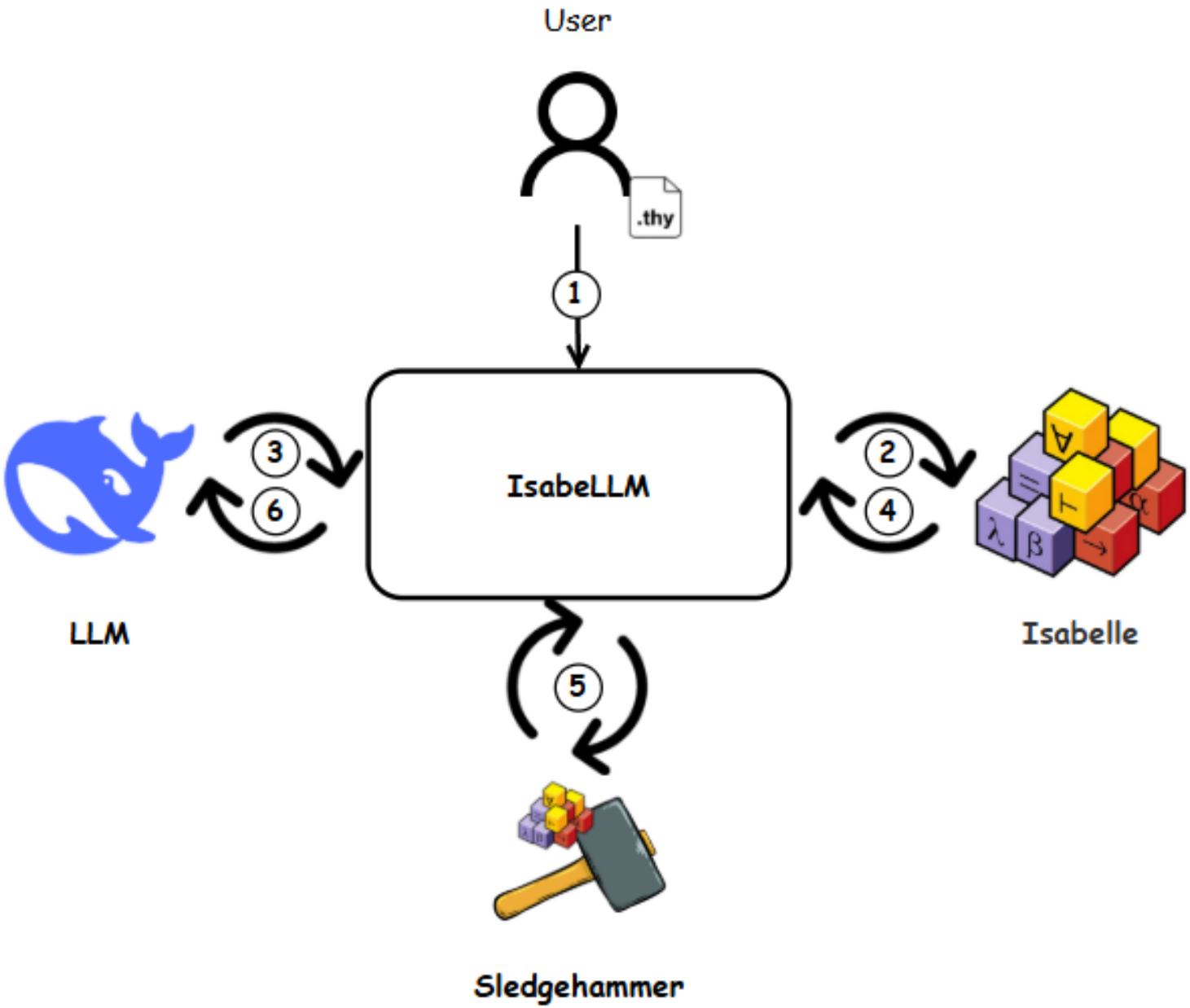
- Formal DeFi components [?, ?, ?]
- KEVM [?], Certora [?], Mythril [?]



Model

Lemma Name	Binary Tree	N-ary Tree
subtree_height	N/A	15
height_mono	1+1	23
obtain_max	N/A	23
foldr_max_eq	N/A	37
branch_height	N/A	30
sub_longest	N/A	28
sub_branch	N/A	41
weaken_distance	1	18
weaken_depth	1	15
common_prefix	25+12	38
height_add (mining)	10+5	36
check_add (mining)	49+158	1
height_add (honest)	10+5	32
check_add (honest)	22+13	36
bounded_check	56	17
consensus	1	5

IsabeLLM



Results

To test the effectiveness of IsabeLLM, we try to prove each of the 16 lemmas listed in Table **??** 10 times with a maximum of 5 iterations per attempt, not counting instances when the LLM would return an empty response. It should be noted that the LoP specified for each lemma can vary as the LLM can generate different proofs for the same thing. As mentioned previously, we used the DeepSeek R1 API for this experiment. Generally speaking, IsabeLLM was able to prove each lemma multiple times, often with a varying number of iterations required to do so. Some lemmas, like `subtree_height`, were repeatedly solved with one iteration but almost always required intervention to amend either incorrect syntax or Sledgehammer incorrect proof steps.
As expected, we tended to see fewer successful attempts for the larger proofs. The lemmas with which it seemed to struggle most was `branch_height` and `bounded_check`.

Discussion

- **API Limitations:** Speed and reliability issues with DeepSeek R1 free API on OpenRouter; occasional empty outputs required retry logic
- **LLM Hallucinations:** Common occurrences of incorrect Isabelle syntax, non-existent theorems, and impossible proofs; manual intervention sometimes needed
- **Sledgehammer Efficiency:** Running on Isabelle2022 prevents access to improved versions; remote calls lack counterexample generation; repeated calls on identical proof steps waste computation
- **Memory Issues:** High memory consumption from back-to-back Sledgehammer calls causes process termination
- **Future Improvements:** Upgrade to later Isabelle releases, implement repeated step detection with caching, improve LLM and Sledgehammer performance over time

.thy

```
lemma subtree_height:
  assumes "t ∈ set ts"
  shows "foldr max (map nHeight ts) 0 ≥ nHeight t"
  sorry
```

Conclusion

In this paper, we introduce the proof automation tool **IsabeLLM** for Isabelle proof assistant. We then used IsabeLLM to complete a novel verification of PoW consensus and analysed its effectiveness.

An area of future work would be to modify IsabeLLM so that it constructs a proof tree by querying the LLM in parallel and branching the proof in different directions for each different proof the LLM gives. This is the standard method used in the field for AI for theorem proving [**?**, **?**] and would help prevent IsabeLLM from getting stuck in a loop of repeatedly trying to prove the same step. This could be taken further by using different LLMs, which would likely generate different approaches to the proof. As the field progresses, more advanced models like Claude Opus 4 are likely to replace our choice of DeepSeek.

Another area of work is using IsabeLLM for more complex proofs that are not necessarily within the blockchain domain and split across multiple theory files. As mentioned previously, IsabeLLM is designed for general purpose and so can be used for proofs in any domain. Furthermore, LLMs could also be fine-tuned on proof corpora datasets like the Archive of