UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Alice Cooper

Type Inference for Fourth Order Logic Formulae

Master's Thesis (30 ECTS)

Supervisor(s): Axel Rose, MSc
May Flower, PhD

Tartu 2023

Type Inference for Fourth Order Logic Formulae

Abstract:
Many interpreting program languages are dynamically typed, such as Visual Basic or Python. As a result, it is easy to write programs that crash due to mismatches of provided and expected data types. One possible solution to this problem is automatic type derivation during compilation. In this work, we consider study how to detect type errors in the WHITESPACE language by using fourth order logic formulae as annotations. The main result of this thesis is a new triple-exponential type inference algorithm for the fourth order logic formulae. This is a significant advancement as the question whether there exists such an algorithm was an open question. All previous attempts to solve the problem lead lead to logical inconsistencies or required tedious user interaction in terms of interpretative dance. Although the resulting algorithm is slightly inefficient, it can be used to detect obscure programming bugs in the WHITESPACE language. The latter significantly improves productivity. Our practical experiments showed that productivity is comparable to average Java programmer. From a theoretical viewpoint, the result is only a small advancement in rigorous treatment of higher order logic formulae. The results obtained by us do not generalise to formulae with the fifth or higher order.

Keywords:

List of keywords

CERCS:
CERCS code and name: `https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e`

Tüübituletus neljandat järku loogikavalemitele

Lühikokkuvõte:

One or two sentences providing a basic introduction to the field, comprehensible to a scientist in any discipline.

Two to three sentences of more detailed background, comprehensible to scientists in related disciplines.

One sentence clearly stating the general problem being addressed by this particular study.

One sentence summarising the main result (with the words “here we show´´ or their equivalent).

Two or three sentences explaining what the main result reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.

One or two sentences to put the results into a more general context.

Two or three sentences to provide a broader perspective, readily comprehensible to a scientist in any discipline, may be included in the first paragraph if the editor considers that the accessibility of the paper is significantly enhanced by their inclusion.

Võtmesõnad:

List of keywords

CERCS:

CERCS kood ja nimetus: `https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e`

Contents

Unsolved issues

1 Introduction

One is where and when one claims to be - this is the underlying principle of most of today's location-based services. This principle, however, hides a whole set of preceding premises that implicitly assert trust in the subject's honesty in reporting its correct location. Having this trust delegated, the reliance on a trusted third party, frequently an atomic computing entity, is still subject to tampering, repudiation, inaccuracy, punctual and single failure, or any other kind of Byzantine behaviour.

The trust levels required to testify to one's alibi remain unmeasurable in modern arrangements. Strategic interactions between rational agents often support this trust. One party provides a location-based service, and another party makes use of it for its individual benefit and by providing a non-tampered time-conscious piece of location claim. This interaction appears to be one of a non-zero-sum game that can be observed in most GPS-based services, mapping platforms, navigation systems, mobility and ride-hailing apps, among many others. If driven by the reasoning goal of extracting correct information from the interacting system, users are logically motivated to report an accurate location. The services, having the higher goal of not losing users due to their reported malfunctioning or inaccuracy, are thus motivated to provide maximized quality when operating and consuming the location claims.

This paradigm is now ubiquitous, which may lead to its fallacious use in other very distinct scenarios. Those scenarios are, therefore and inversely, the ones that fundamentally require verifiable proof of location to assert a particular state or derive a conclusion. Consider, for example, scenarios requiring location-based authentication or authorization in adversarial environments that rely on information gathered in a trustless setup. These materialize into services requiring, for instance, a digital certificate as proof that a given user is within a particular geographical area, to enable certain functionalities or assert liability, as in location-based access control, review or reward systems, augmented reality games, social networks, etc... Security against geo-tampering or location spoofing in a relatively trustless environment is needed to achieve the required integrity.

The basic infrastructural concept is somewhat understood, and theoretical or experimental solutions have been delivered throughout the years. These solutions have evolved parallel with their trust assumptions, beginning with a fully trusted setup and progressively shifting towards modern requirements for operational decentralization - of power and profit. Most recent attempts contemplate the need for a permissionless means of reaching consensus between a quorum of witnesses that can attest to one's presence at a given point in space and at a given moment in time. These concepts take shape with a combination of tools: wireless technologies as message-exchanging means, cryptographic protocols as confidentiality, integrity, or authentication enablers, and distributed ledgers as publicly trusted record keepers.

The quest for a solution that could make these location-based services as prevalent and ubiquitous shall aim to address a set of design challenges. These challenges are, among others, the solution's flexibility and deployability, preferably by making use of existing infrastructure, or at least accessible technology, and the solution's security and privacy, obeying the modern cryptographic standards and requirements, to guarantee some level of privacy, and resiliency to attacks. This thesis, aiming to address these matters, delivers the following contributions:

1. A semi-formalization of the location-based services paradigm, including the underlying premises and the strategic interactions between rational agents, along with a review of the state of the art in the field. The review is discriminated in terms of trust levels, from fully trusted to permissionless environments, and in terms of the underlying technology, from centralized to decentralized.

2. The design and implementation of a proof-of-concept that can be deployed in a permissionless manner, using existing infrastructure, and that can be used to attest to one's presence at a given point in space and time. Specifically, the proof-of-concept is based on the use of routing protocols for multi-hop mobile ad hoc networks to set up a mesh network of witnesses that can attest to one's presence in a given geographical area.

The structure of the work is as follows. In chapter 2, an introduction to the underlying concepts and hypotheses is provided, as well as a mention to the technology involved in the practical implementation. Chapter 3 examines and analyzes similar work discriminated in terms of trust levels. In chapter 4, a general overview of the requirements for the proposed solution is given. Chapter 5 details the architecture's design, implementation, and evaluation. Finally, chapter 6 presents the conclusion and recommendations for future work.

2 Prelude

This section introduces not only the underlying concepts that sustent the work, but also the technology involved in implementing the proposed proof of concept.

2.1 Proof-of-Location

2.1.1 Parties Involved

The general act of witnessing alludes to the simultaneous spatiotemporal existence of a set of entities with distinct roles. The majority of the protocols convey a clear distinction between these roles, highlighting the relative dynamism that distinguishes those entities.

In comparable terms, highly dynamic entities do not maintain a fixed geographical location for long periods of time. They are often observed in movement, thereby repeatedly starting and finishing communication procedures with nearby entities. On the other hand, static entities are expected not to engage in frequent position changes, expressing continuous and fairly invariable communication availability around a fixed point in space as time passes [1]. The act is, however, only completed with another type of entity from whom neither the relative staticity nor the relative dynamism frankly matters. These protocol parties are often external and asynchronous to the witnessing process, but they do effectively take a non-negligible part in incentivizing and giving significance to the witnessing act.

Concisely and in concrete terms, these location-proof arrangements expect the existence of a *prover* that engages in any communication protocol with nearby participants, the *witnesses*, with the goal of gathering a verifiable proof-of-location claim, to be later presented to a *verifier*, therefore convincing it of one's existence within a geographical area at a given moment in the past [2].

Prover. A prover is a dynamic entity, both in movement and availability terms, that is expected to be able to communicate with the witnesses to gather a proof of its location and to be later able to provide a location claim to the verifier. Communication with nearby witnesses is thought to happen wirelessly, using any short-range message transmission means. Provers are also expected to be associated with a verifiable but desirably private identity, often as a pseudonym.

Witness. A witness is adjunctly an entity that is expected to be able to communicate with the prover via the same short-range communication channel and to be able to provide it with a verifiable piece of location attestation. The witnesses are envisioned to seldomly change their absolute location and maintain a relatively stable neighbouring list of nearby witnesses. These references aim at attaining the figurative creation of coverage zones as strongly connected graphs that form the boundaries of the atomic units of a polygonal mesh. Witnesses are as well expected to be identified, usually by a pseudonym.

Verifier. A verifier is an external entity that is expected to be able to receive a location claim from a prover and to be able to verify its validity. Even though possible and predicted for trusted setups, in a trustless environment and with the general assurances of a permissionless protocol, verifiers shall not have the need to communicate directly with the witnesses. Verifiers' identity is also of no measurable importance for the protocol, as the interaction between the prover and the verifier is usually asynchronous and external to the witnessing process.

2.1.2 Adversary Models

2.2 Wireless Mesh Networks

The envisioned fourth industrial revolution has set the track for modern advancements in achieving a global web of pervasive connectivity between all sorts of machines [3, 4]. New means of radio and wireless communication have been pushing for the technological heterogeneity of protocols, architectures, devices, and consequent performance levels in order to find their design suitability for different coverage or range scenarios, transmission or bandwidth rates [5]. Additionally, requirements for more complex, adaptable, and resilient topologies have captured broad interest, in both academic and industry domains [4].

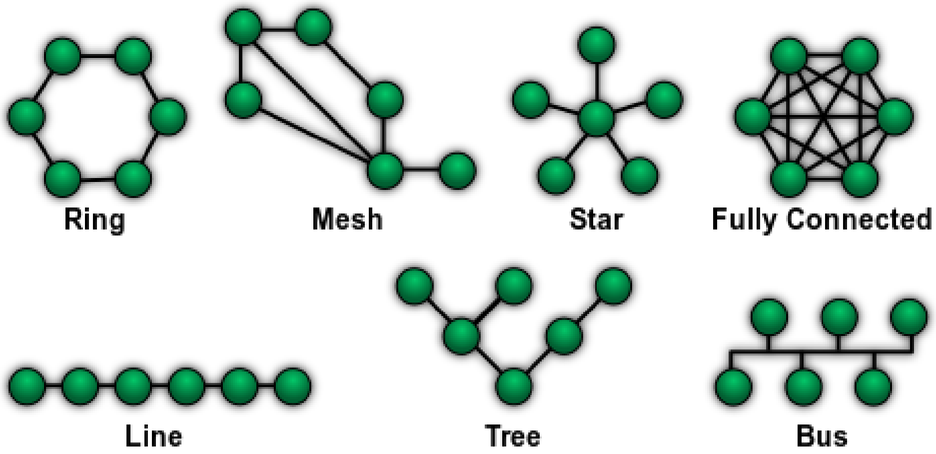


Figure 1. Different topologies of a computer network.

Draw my own image for network topologies.

The development of new hardware architectures, protocols and applications started gaining momentum and branched their way forward to support the rising popularisation of Wireless Mesh Networks (WMNs). In mesh topologies (see Fig. 1), network nodes are directly and dynamically connected in a non-hierarchical way. This trait eventually allows for many-to-many communications between the devices to efficiently route data from a generic source to a generic destination. The infrastructure nodes that make up the mesh are expected to dynamically self-organize and configure themselves, resulting in beneficial distributed effects on the overall fault tolerance, ease of deployment, and workload allocation [4, 5]. WMNs follow these principles with the particularity of being made up of radio nodes that communicate via any sort of wireless technology.

Some of the most common wireless technologies that have been, throughout the years, ported to WMNs are IEEE 802.11, Bluetooth, IEEE 802.15, and LoRa. The first is the most popular and widely used, being the basis for the Wi-Fi standard, which, at the beginning of the last decade, saw an amendment that mainly targeted mesh networks, the IEEE 802.11s WLAN Mesh Standard [6]. The novelty came with the introduction of routing mechanisms operating at the ISO/OSI Layer 2, allowing for compatible information delivery in the layers above. The dynamic establishment of a topology for IEEE 802.11s-based mesh networks relies on the phased transmission of beacon messages that allow for the discovery, synchronization, and maintenance of the links between the peers. IEEE 802.11s has a default routing protocol, the Hybrid Wireless Mesh Protocol (HWMP), which is based on a series of flooding procedures for both proactive and reactive path finding and selection [7]. This protocol is, however, not strictly enforced by the standard and has been replaced by other, more popular solutions. One notable example is the Better Approach To Mobile Ad-hoc Networks (B.A.T.M.A.N.) routing protocol.

This thesis will explore the concept of WMNs and their potential for serving as the infrastructural topology that enables the relatively short-ranged wireless exchange of messages between the participants of a proof-of-location protocol. The following sections will present the B.A.T.M.A.N. routing protocol, OpenWRT and other relevant tools that will be later used to implement the proof of concept.

2.2.1 B.A.T.M.A.N. Routing Protocol

The Better Approach To Mobile Ad-hoc Networks (B.A.T.M.A.N.)[1] is a proactive routing protocol for WMNs that operates not at the network layer but at the data link layer, by asserting the reliability of radio links using routing metrics and a distance-vector approach [8]. Its newer wireless version, *batman-adv*, has gained traction and popularity and eventually made itself available in the Linux kernel.

[1]http://www.open-mesh.org/

Route discovery is preemptively replaced with neighbour discovery, and each infrastructural node is instructed to calculate its potential best next-hop, significantly reducing the overhead of requiring each peer to be aware of the whole network topology. Its version V introduced a throughput metric to evaluate the links' quality and routing choices, replacing the version IV packet loss-based metric, deemed unsuitable for larger network sizes [8].

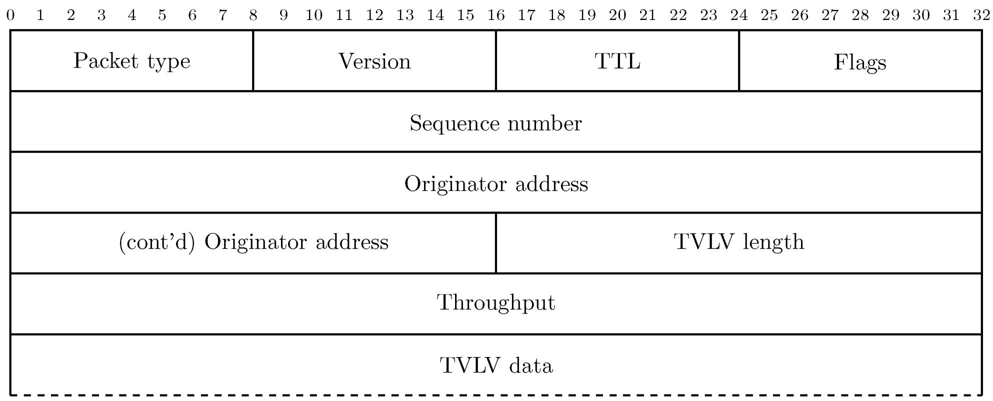


Figure 2. OriGinator Message version 2 (OGMv2) packet format [4, 9].

The discovery of neighbouring nodes is accomplished by broadcasting OriGinator Messages (OGMv2, see Fig. 2), featuring a collision avoidance delay mechanism, the detection of new or duplicate messages, and other fields for throughput measurement and gateway discovery [4]. Hence, the OGM flooding protocol enables mesh routing procedures that, simultaneously but independently, allow for estimating the quality of the individual links. Additionally, the protocol enables OGM aggregations as an effort to reduce the overhead of sending many short-sized frames. Nevertheless, there is still a quest for optimizations that would allow for more efficient use of multiple interfaces. An implementation of a subset of the Internet Control Message Protocol (ICMP) was also made available, allowing, for instance, the use of the *ping* command to test the connectivity between nodes [8].

Building on the previous, the B.A.T.M.A.N. routing protocol has been, through multiple initiatives, successfully blended into the OpenWRT project, which will also be employed in the proof of concept. The following section will present OpenWRT and other relatable tools.

2.2.2 OpenWRT, QEMU, and Raspberry Pis

The OpenWRT project [2] is a Linux distribution for embedded devices, which, in the context of this thesis, will serve as the host operating system for running the proof of concept solution. The project is based on the Linux kernel, encapsulating several of its libraries and packages, and is designed to be used on devices with limited resources. OpenWRT features not only a modular architecture powered by a writable root filesystem and automation build tools with integrated cross-compiler toolchain, but also a package management system that allows for the installation of additional software. The project also provides extensive configuration options for networking capabilities, including enabling mesh networking support through the B.A.T.M.A.N. routing protocol.

To facilitate the development and testing of the proof of concept, the QEMU[3] emulator will be used. QEMU is a generic and open-source machine virtualizer that, through its versatile set of features, allows for the full-system emulation of a wide range of hardware and software. The emulator will run the OpenWRT-generated images and spawn multiple virtual machines. These machines will simulate the various protocol participants by establishing a mesh network with the help of the virtualizer network emulation tools. The intention is to ease and accelerate the development process by allowing for testing the proof of concept in a controlled environment, without the management, maintenance and deployment hustle of physical devices. Later, the solution will eventually be deployed on a set of Raspberry Pis[4], the most widely used single-board computers for developing IoT solutions.

2.3 Permissionless Consensus

Long has been the time when consensus was still on the verge of being considered such a fundamental problem of distributed systems, at least as widely as it is now [10, 11, 12]. Generally, consensus means reaching an agreement between multiple parties in the potential presence of faulty individuals. As per multi-agent systems, interacting over computer networks, consensus is thought to be the result of a coordination effort, such that those parties agree on some value at a given moment. Achieving consensus implies that the system shall be reliable and fault-tolerant. However, the consensus problem has been invariably and evolutionarily limited by some assumptions on the networks and on the roles of its participants. The well-known "secure Byzantine-Fault-Tolerant multiparty consensus systems" that have been designed over the years are usually meant to work only with a set of known participants, faulty or not [13]. The other side of the coin is the permissionless consensus challenge, consisting of achieving agreement in an environment where the parties are unknown and untrusted [14, 15]. There are other

[2] https://openwrt.org/
[3] https://www.qemu.org/
[4] https://www.raspberrypi.org/

intrinsic particularities of this type of networks, for example, their openness, and the lack of any kind of central authority, that consequently add another layer of complexity to the problem. The participants are not only unknown and untrusted but can also join or leave the network at any time, freely choosing if they want to participate in the consensus protocol or not. Nevertheless, the problem of permissionless consensus can still be seen as a special case of the more general consensus and can still be formalized in a similar way.

The first theoretical algorithms were designed for synchronous systems, where the communication between the processes is reliable, and the delay is bounded. The first practical asynchronous consensus algorithm was later proposed by Castro and Liskov in [13]. And naturally, after that work, many other asynchronous consensus algorithms have appeared. However, all of them are based on the assumption that the number of faulty processes is less than a certain threshold. Additionally, the assumption of a known set of participants is also made, as well as their roles in the consensus protocol. These are very strong assumptions that limit the challenges that can be addressed, for instance, with the impossibility of knowing the participants beforehand as they may participate anonymously, or dynamically.

2.3.1 Proof-of-Work and Proof-of-Stake

The advancements of the internet more than potentiated the revolution and what we now call the permissionless consensus problem was finally born. Without forgetting the previous attempts, the first practical permissionless consensus algorithm was proposed by Nakamoto in [14]. It is a proof-of-work consensus protocol that resembles a "replicated state machine" where the independent participants reach agreement not only about transactional values, but also about their order - naturally forming the underlying structure of what is now known as a blockchain. "Proof-of-work is essentially one-CPU-one-vote" and this is the novelty introduced by Bitcoin [16, 17]. The focus shifted for decentralized systems and after proof-of-work many other consensus mechanisms have been proposed, based on different consensus units, being them proof-of-stake, proof-of-space, proof-of-authority, and a chaotic diversity of related ones.

The solution is, nevertheless, not settled and the scientific community has been reasoning about the need for permissionless consensus when there are already well known and established consensus protocols that work in trusted environments [13, 18]. However, even those protocols have their own limitations, not only in terms of trust, fault-tolerance, centrality, permissions, or bottlenecks, but also in terms of scalability [18]. Trying to put some effort on decentralization, Byzantine-Fault-Tolerant consensus protocols, or simply BFT protocols, are not known for their performance when the network grows in size. The more participants there are, the more messages need to be exchanged between them, and the more time it takes to reach consensus, even if assuring deterministic finality [19]. This is a problem that is not only related to the number of participants, but also to the communication fashion and bandwidth, and the computational capacity of the devices that participate in the consensus protocol. Summing up, the need for permissionless consensus is then justified by the fact that permissioned protocols are not compatible with the requirements of the new generation of distributed systems, especially in the context of blockchain networks. These requirements include dealing with today's sparse networks of anonymously and dynamically participating devices, without interrupting consensus and while battling Sybil attacks [20, 21]. Fundamentally, the permissionless consensus problem is the need for a consensus protocol that can be run in a distributed and decentralized environment, where the participants are unknown and untrusted, and where the network is bigger, sparser and unpredictably less reliable.

2.3.2 Proof-of-X

2.4 Title of Subsection 2

Rule: If you divide the text into subsections (or subsubsections) then there has to be at least two of them, otherwise do not create any.

Tip: You can also use paragraphs, e.g.

Type rules for integers. Some text ...

Type rules for rational numbers. Some text here too...

2.5 How to use references

Cross-references to figures, tables and other document elements. LaTeX internally numbers all kind of objects that have sequence numbers:

- chapters, sections, subsections;
- figures, tables, algorithms;
- equations, equation arrays.

To reference them automatically, you have to generate a label using `\label{some-name}` just after the object that has the number inside. Usually, labels of different objects are split into different namespaces by adding dedicated prefix, such as `sec:`, `fig:`. To use the corresponding reference, you must use command `\ref` or `\eqref`. For instance, we can reference this subsection by calling Section 2.5. Note that there should be a nonbreakable space ~ between the name of the object and the reference so that they would not appear on different lines (does not work in Estonian).

Citations. Usually, you also want to reference articles, webpages, tools or programs or books. For that you should use citations and references. The system is similar to the cross-referencing system in LaTeX. For each reference you must assign a unique label. Again, there are many naming schemes for labels. However, as you have a short document anything works. To reference to a particular source you must use `\cite{label}` or `\cite[page]{label}`.

References themselves can be part of a LaTeX source file. For that you need to define a bibliography section. However, this approach is really uncommon. It is much more easier to use BibTeX to synthesise the right reference form for you. For that you must use two commands in the LaTeX source

- \bibliographystyle{alpha} or \bibliographystyle{plain}
- \bibliography{file-name}

The first command determines whether the references are numbered by letter-number combinations or by cryptic numbers. It is more common to use `alpha` style. The second command determines the file containing the bibliographic entries. The file should end with `bib` extension. Each reference there is in specific form. The simplest way to avoid all technicalities is to use graphical frontend Jabref (`http://jabref.sourceforge.net/`) to manage references. Another alternative is to use DBLP database of references and copy BibTeX entries directly form there.

The following paragraph shows how references can be used. Game-based proving is a way to analyse security of a cryptographic protocol [**?**, **?**]. There are automatic provers, such as CertiCrypt [**?**] and ProVerif [**?**].

3 How to add figures and pictures to your thesis

Here are a few examples of how to add figures or pictures to your thesis (see Figures 3, 4, 5).

Rule: All the figures, tables and extras in the thesis have to be referred to somewhere in the text.

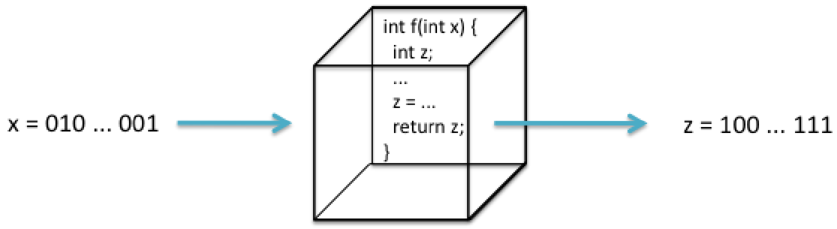


Figure 3. The title of the Figure.

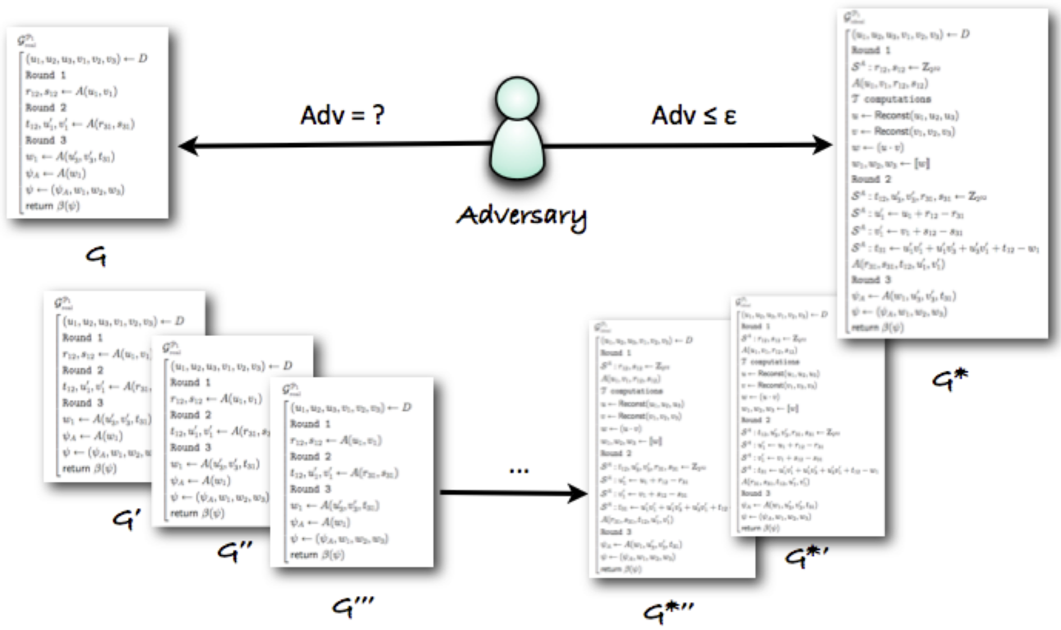


Figure 4. Refer if the figure is not yours [?].

Tip: If you add a screenshot then labeling the parts might help make the text more understandable (panel C vs bottom left part), e.g.

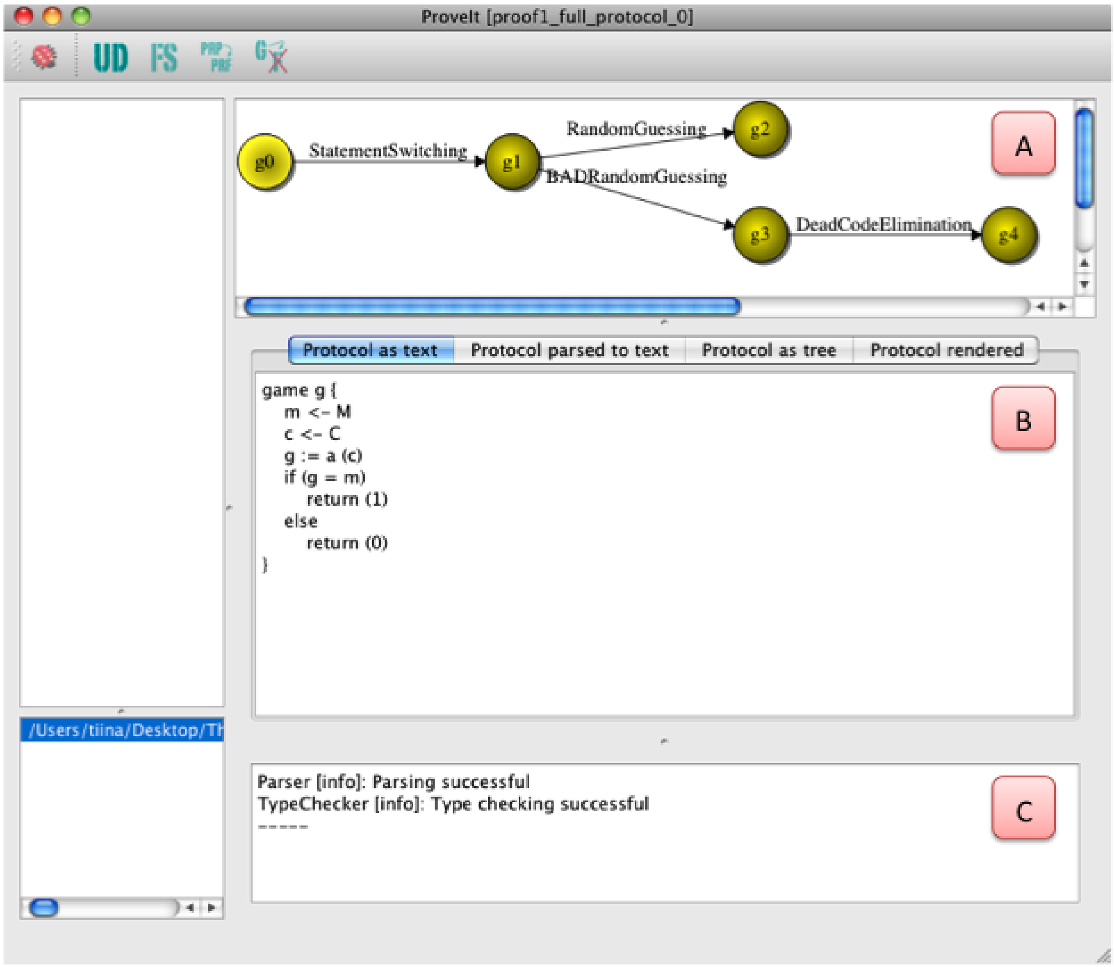
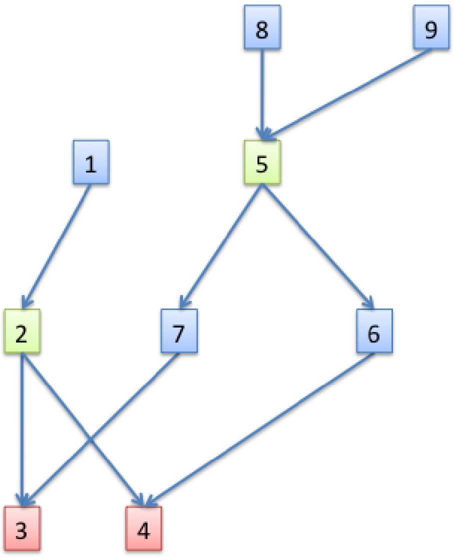


Figure 5. Screenshot of ProveIt.



Node	Decendants
1	2, 3, 4
2	3, 4
3	
4	
5	3, 4, 6, 7
6	4
7	3
8	3, 4, 5, 6, 7
9	3, 4, 5, 6, 7

Figure 6. Example how to put two figures parallel to each other.

Example: A screenshot of ProveIt can be seen on Figure 5. The user first enters the pseudocode of the initial game in panel B. ProveIt also keeps track of all the previous games showing the progress on a graph seen in panel A.

There are two figures side by side on Figure 6.

4 Other Ways to Represent Data

4.1 Tables

Table 1. Statements in the ProveIt language.

Statement	**Typeset Example**
assignment	$a := 5 + b$
uniform choice	$m < -M$
function signature	$f : K \times M- > L$

4.2 Lists

Numbered list example:

1. item one;
2. item two;
3. item three.

4.3 Math mode

Example:

$$a + b = c + d \tag{1}$$

Aligning:

$$\begin{aligned} a &= 5 \\ b + c &= a \\ a - 2 * 3 &= 5/4 \end{aligned}$$

Hint: Variables or equations in text are separated with $ sign, e.g. a, $x - y$.

4.4 algorithm2e

4.5 Pseudocode

4.6 Frame Around Information

Tip: We can use minipage to create a frame around some important information.

Algorithm 1: typeChecking

Input: Abstract syntax tree

Result: Type checking result; In addition, type table $\mathsf{type}_{\mathsf{type\_G}}$ for global variables, $\mathsf{type}_{\mathsf{game}}$ for the main game and $\mathsf{type}_{\mathsf{fun}}$ for each $fun \in F$

```
while something changed in last cycle do
    foreach global statement s do parseStatement(s, type_type_G);
    ;
    foreach function fun do
        foreach statement s in fun do parseStatement(s, type_fun);
        ;
    foreach statement s in game do parseStatement(s, type_game);
    ;
```

```
expression
  : NUMBER
  | VARIABLE
  | '+' expression
  | expression '+' expression
  | expression '*' expression
  | function_name '(' parameters ')'
  | '(' expression ')'
```

Figure 7. Grammar of arithmetic expressions.

1. integer division (\div) – only usable between Int types
2. remainder (%) – only usable between Int types

Figure 8. Arithmetic operations in ProveIt revisited.

5 Conclusion

what did you do?

What are the results?

future work?

References

[1] B. Nasrulin, M. Muzammal, and Q. Qu, "A robust spatio-temporal verification protocol for blockchain," in *Web Information Systems Engineering–WISE 2018: 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part I 19*, pp. 52–67, Springer International Publishing, 2018.

[2] A. Dupin, J.-M. Robert, and C. Bidan, "Location-proof system based on secure multi-party computations," in *Provable Security: 12th International Conference, ProvSec 2018, Jeju, South Korea, October 25-28, 2018, Proceedings*, pp. 22–39, Springer, 2018.

[3] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer networks*, vol. 47, no. 4, pp. 445–487, 2005.

[4] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, "Wireless mesh networking: An iot-oriented perspective survey on relevant technologies," *Future Internet*, vol. 11, no. 4, p. 99, 2019.

[5] M. L. Sichitiu, "Wireless mesh networks: opportunities and challenges," in *Proceedings of World Wireless Congress*, vol. 2, p. 21, 2005.

[6] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "Ieee 802.11 s: the wlan mesh standard," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, 2010.

[7] S. Bari, F. Anwar, and M. Masud, "Performance study of hybrid wireless mesh protocol (hwmp) for ieee 802.11 s wlan mesh networks," in *2012 international conference on computer and communication engineering (ICCCE)*, pp. 712–716, IEEE, 2012.

[8] D. Seither, A. König, and M. Hollick, "Routing performance of wireless mesh networks: A practical evaluation of batman advanced," in *2011 IEEE 36th Conference on Local Computer Networks*, pp. 897–904, IEEE, 2011.

[9] "Open-mesh. originator message version 2 (ogmv2)." `https://www.open-mesh.org/projects/batman-adv/wiki/OGMv2`. Accessed: 2023-02-16.

[10] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 27, no. 2, pp. 228–234, 1980.

[11] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the works of leslie lamport*, pp. 203–226, Springer, 2019.

[12] L. Lamport, “The weak byzantine generals problem,” *Journal of the ACM (JACM)*, vol. 30, no. 3, pp. 668–676, 1983.

[13] M. Castro, B. Liskov, *et al.*, “Practical byzantine fault tolerance,” in *OsDI*, pp. 173–186, 1999.

[14] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” *Decentralized Business Review*, p. 21260, 2008.

[15] V. Buterin *et al.*, “A next-generation smart contract and decentralized application platform,” *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[16] R. Pass and E. Shi, “Hybrid consensus: Efficient consensus in the permissionless model,” *Cryptology ePrint Archive*, 2016.

[17] R. Pass and E. Shi, “Hybrid consensus: Scalable permissionless consensus,” 2016.

[18] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, “The honey badger of bft protocols,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 31–42, 2016.

[19] C. Decker, J. Seidel, and R. Wattenhofer, “Bitcoin meets strong consistency,” in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, pp. 1–10, 2016.

[20] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, “A survey on consensus mechanisms and mining strategy management in blockchain networks,” *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[21] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, “A survey of distributed consensus protocols for blockchain networks,” *IEEE Communications Surveys and Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

Appendix

I. Glossary

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Alice Cooper**,
(author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Type Inference for Fourth Order Logic Formulae,
   (title of thesis)

   supervised by Axel Rose and May Flower.
   (supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Alice Cooper
dd/mm/yyyy