

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 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

1	Introduction	6
2	Title of Section 2	8
2.1	Title of Subsection 1	8
2.1.1	Title of Subsubsection 1	8
2.1.2	Title of Subsubsection 2	8
2.2	Title of Subsection 2	8
2.3	How to use references	8
3	How to add figures and pictures to your thesis	10
4	Other Ways to Represent Data	13
4.1	Tables	13
4.2	Lists	13
4.3	Math mode	13
4.4	algorithm2e	13
4.5	Pseudocode	13
4.6	Frame Around Information	13
5	Conclusion	15
	References	16
	Appendix	17
	I. Glossary	17
	II. Licence	18

Unsolved issues

List of keywords	2
CERCS code and name: https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e	2
One or two sentences providing a basic introduction to the field, comprehensible to a scientist in any discipline.	2
Two to three sentences of more detailed background, comprehensible to scientists in related disciplines.	2
One sentence clearly stating the general problem being addressed by this particular study.	2
One sentence summarising the main result (with the words “here we show” or their equivalent).	2
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.	2
One or two sentences to put the results into a more general context.	3
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.	3
List of keywords	3
CERCS kood ja nimetus: https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e	3
What is it in simple terms (title)?	6
Why should anyone care?	6
What was my contribution?	6
What you are doing in each section (a sentence or two per section)	6
Short description of what this section is about	8
what did you do?	15
What are the results?	15
future work?	15

1 Introduction

What is it in simple terms (title)?

Why should anyone care?

What was my contribution?

What you are doing in each section (a sentence or two per section)

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 forcibly assert fundamental honesty and implicit trust in the subject for 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 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, such as mapping platforms, navigation systems, mobility and ride-hailing services, 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 unlock a particular outcome or derive a conclusion. Consider scenarios requiring location-based authentication or authorization, adversarial environments that need to rely on information gathered in a trustless setup, and many others. 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, or even reward systems, augmented reality games, or social networks. Security against geo-tampering or location spoofing in a relatively trustless environment is needed to achieve the required integrity.

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.
2. A formalization of the location-based services paradigm, including the underlying premises and the strategic interactions between rational agents.

3. A formalization of the location-based services paradigm, including the underlying premises and the strategic interactions between rational agents.

The structure of the work is as follows. In chapter 2, an introduction to the underlying concepts and hypotheses is provided. 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 Title of Section 2

Short description of what this section is about

2.1 Title of Subsection 1

Some text...

2.1.1 Title of Subsubsection 1

Some text...

2.1.2 Title of Subsubsection 2

Some text...

2.2 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...

2.3 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 2.3. 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 from there.

The following paragraph shows how references can be used. Game-based proving is a way to analyse security of a cryptographic protocol [BR04, Sho04]. There are automatic provers, such as CertiCrypt [BGZ09] and ProVerif [Bla].

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 1, 2, 3).

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

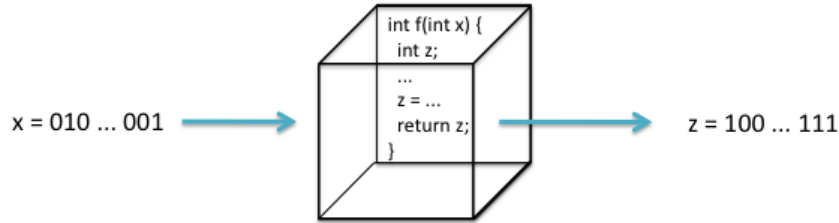


Figure 1. The title of the Figure.

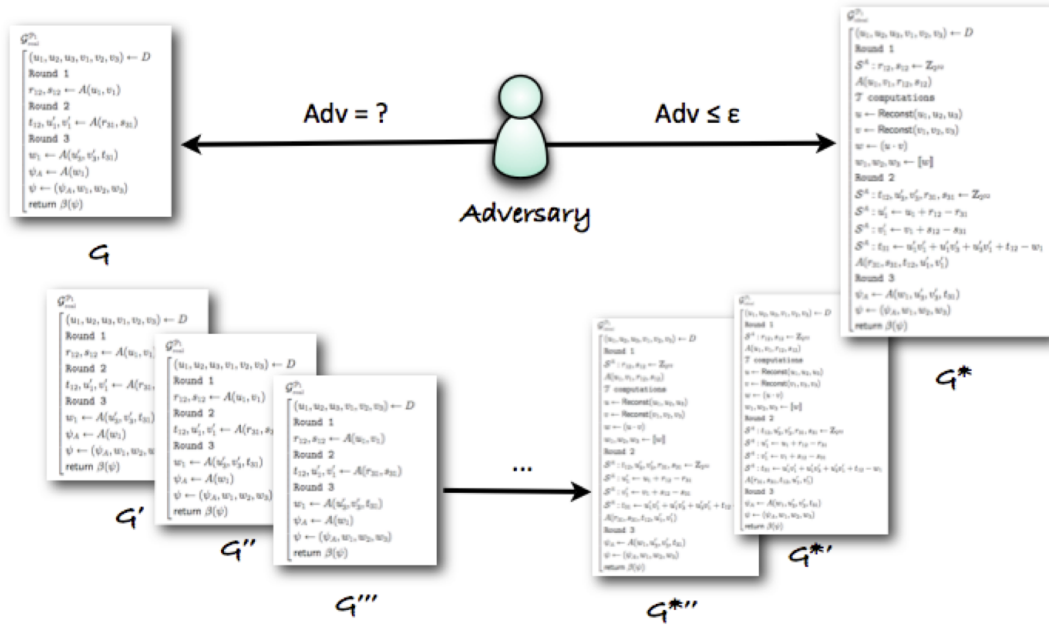


Figure 2. Refer if the figure is not yours [Kam12].

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.

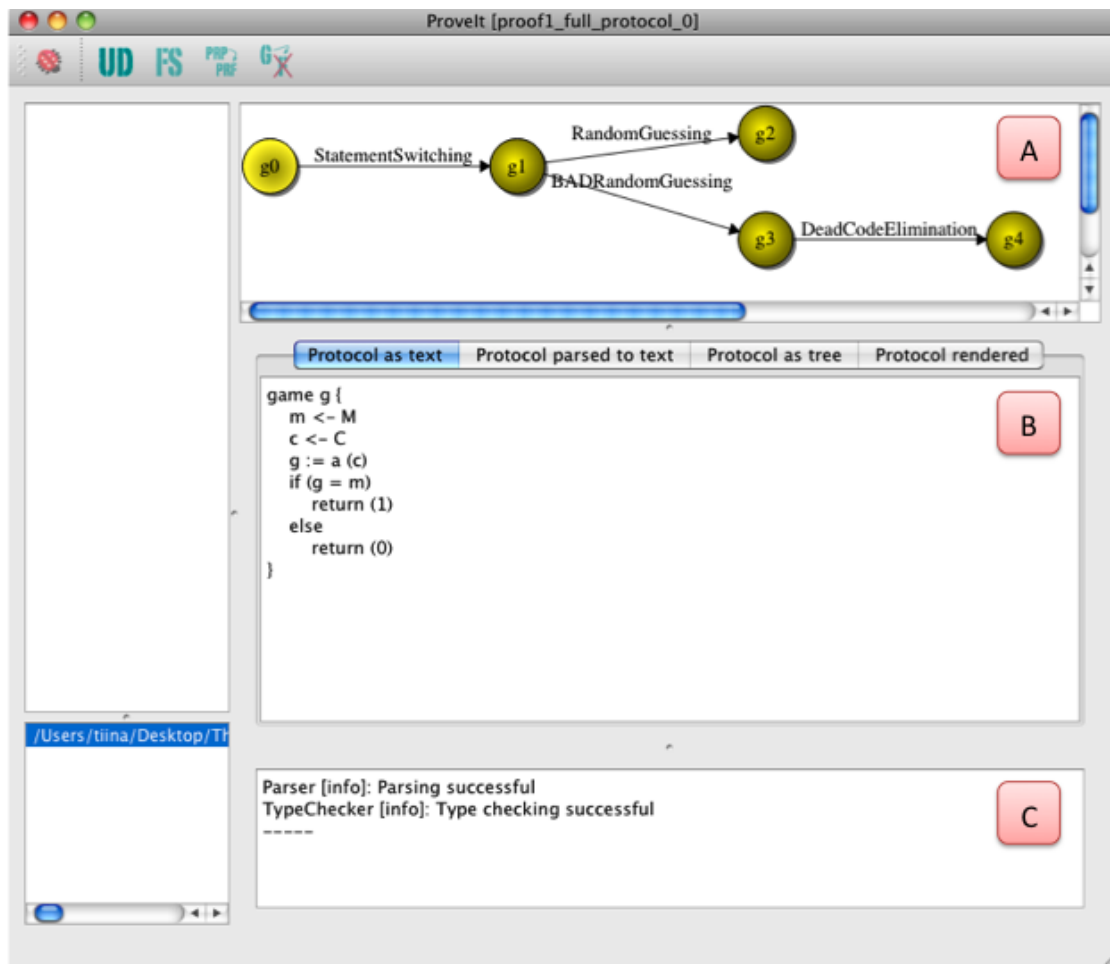


Figure 3. Screenshot of ProveIt.

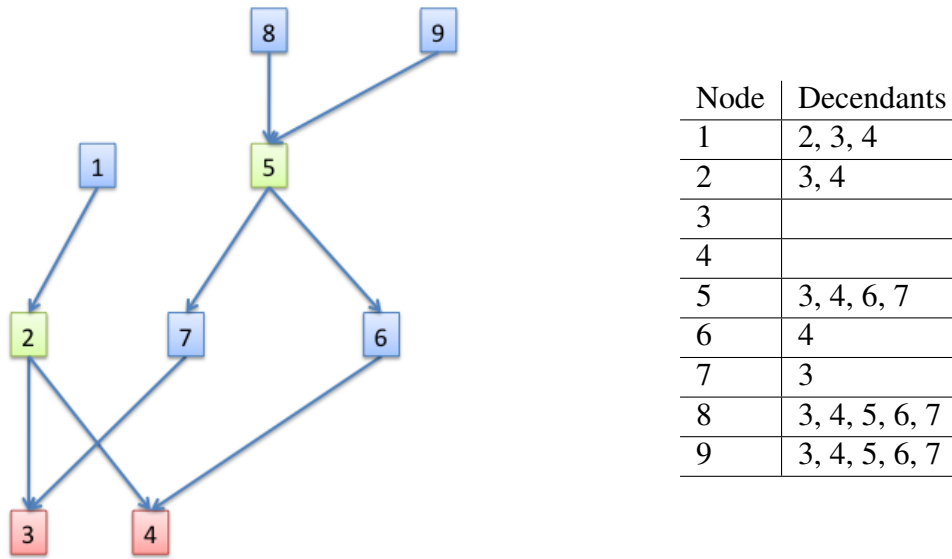


Figure 4. Example how to put two figures parallel to each other.

Example: A screenshot of ProveIt can be seen on Figure 3. 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 4.

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	$f : K \times M \rightarrow L$

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:

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

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 $\text{type}_{\text{type_G}}$ for global variables, $\text{type}_{\text{game}}$ for the main game and type_{fun} for each $\text{fun} \in F$

```
1 while something changed in last cycle do
2   foreach global statement s do parseStatement(s,  $\text{type}_{\text{type\_G}}$ );
3   ;
4   foreach function fun do
5     foreach statement s in fun do parseStatement(s,  $\text{type}_{\text{fun}}$ );
6   ;
7   foreach statement s in game do parseStatement(s,  $\text{type}_{\text{game}}$ );
8   ;
```

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

Figure 5. Grammar of arithmetic expressions.

- | |
|---|
| <ol style="list-style-type: none">1. integer division (<code>\div</code>) – only usable between <code>Int</code> types2. remainder (<code>%</code>) – only usable between <code>Int</code> types |
|---|

Figure 6. Arithmetic operations in ProveIt revisited.

5 Conclusion

what did you do?

What are the results?

future work?

References

- [BGZ09] Gilles Barthe, Benjamin Grégoire, and Santiago Zanella Béguelin. Formal certification of code-based cryptographic proofs. In *36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2009*, pages 90–101. ACM, 2009.
- [Bla] Bruno Blanchet. Proverif: Cryptographic protocol verifier in the formal model. <http://www.proverif.ens.fr/>.
- [BR04] Mihir Bellare and Phillip Rogaway. Code-based game-playing proofs and the security of triple encryption. Cryptology ePrint Archive, Report 2004/331, 2004. <http://eprint.iacr.org/>.
- [Kam12] Liina Kamm. ProveIt – How to make proving cryptographic protocols less tedious. Talk at the 21st Estonian Computer Science Theory Days at Kubija, January 2012.
- [Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. <http://eprint.iacr.org/>.

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