Verification of Chunk Sequences
Research internship proposal, M2

Arthur Charguéraud, Inria, Strasbourg, arthur.chargueraud@inria.fr
François Pottier, Inria, Paris, francois.pottier@inria.fr

2019–2020

1 Background

Since the days of Floyd and Hoare, "program logics", that is to say, deduction systems for proving properties of programs, have made important progress. A decisive step was the introduction by Reynolds of Separation Logic, which emphasizes "local reasoning", that is, modular reasoning in the presence of heap-allocated mutable state.

Today, the CFML system [2], an implementation of Separation Logic for the OCaml programming language, allows carrying out machine-checked proofs of OCaml libraries. For example, CFML has been used to verify data structures (vectors, hash tables, disjoint set forests, finger trees, etc.) and algorithms (Dijkstra's shortest paths algorithm, depth-first search, Eratosthenes' sieve, incremental cycle detection, etc.). These proofs are constructed interactively inside the Coq proof assistant.

2 Summary

The goal of this internship is to verify an efficient implementation of sequences, called "chunk sequences". This implementation uses so-called "chunks", that is, small arrays of elements, which are organized in a tree-shaped data structure [1]. In recent work, Charguéraud and Pottier designed a new version of chunk sequences that offers, at the same time:

- ephemeral sequences (that is, sequences that support in-place mutation),
- persistent sequences (that is, sequences that appear to be immutable, although their implementation can involve mutable state and copy-on-write techniques),
- constant-time operations for converting between these two variants.

This new data structure is already implemented as an OCam library. The library has been extensively tested using "fuzz testing" [5], a form of unit testing. Yet, its correctness is far from obvious. There could well remain correctness or performance bugs, which would be highly problematic if this sequence library is to be widely distributed. Thus, this new chunk sequence data structure is a prime candidate for a program verification effort.

3 Roadmap

Here is a possible roadmap for this internship. It is unlikely that there will be time to deal with all items on this list. Dealing with the first item constitutes a baseline goal. Dealing with the first two items would be remarkable.

1. Gain familiarity with the metatheory and practical use of CFML. Verify an implementation of *ephemeral* sequences.

2. Extend the proof to support for *persistent* sequences and for conversions between ephemeral and persistent sequences. Verify this implementation; in particular, verify the copy-on-write mechanisms at play. To the best of our knowledge, this would be the first formal verification of a persistent data structure implemented with optimized imperative code.

3. Extend the proof to support ad-hoc representations for *short* sequences. (Indeed, sequences with fewer than a couple hundred elements can be represented more efficiently than with a tree of chunks.)

4. Using CFML's "time credits" [3, 4], verify the time complexity of the data structure in the case of ephemeral accesses. Then, attempt to state and prove bounds for particular usage scenarios involving a mixture of ephemeral and persistent acccesses.

4 Prerequisites

Familiarity with the operational semantics of programming languages (MPRI 2.4), with Hoare logic and Separation Logic (MPRI 2.36.1), and with proof assistants (MPRI 2.7.1 and 2.7.2) is essential. A solid programming background, including fluency in OCaml, is also highly desirable.

5 Practical details

This internship will be co-supervised by Arthur Charguéraud and by François Pottier. It will take place from March 2020 to August 2020 approximately. It can take place *either* in Paris or in Strasbourg.

References

[1] Umut A. Acar, Arthur Charguéraud, and Mike Rainey. Theory and practice of chunked sequences. In *Algorithms (ESA)*, volume 8737, pages 25–36. Springer, 2014.

[2] Arthur Charguéraud. The CFML tool and library. `http://www.charglueraud.org/softs/cfml/`, 2019.

[3] Armaël Guéneau, Arthur Charguéraud, and François Pottier. A fistful of dollars: Formalizing asymptotic complexity claims via deductive program verification. In *European Symposium on Programming (ESOP)*, volume 10801 of *Lecture Notes in Computer Science*, pages 533–560. Springer, 2018.

[4] Armaël Guéneau, Jacques-Henri Jourdan, Arthur Charguéraud, and François Pottier. Formal proof and analysis of an incremental cycle detection algorithm. In *Interactive Theorem Proving (ITP)*, volume 141 of *Leibniz International Proceedings in Informatics*, pages 18:1–18:20, 2019.

[5] François Pottier. Testing an implementation of sequences, 2018. `https://gitlab.inria.fr/fpottier/seq-test`.