Hayley Patton

hayley@applied-langua.ge — https://github.com/no-defun-allowed — Melbourne, Australia

PROJECTS

Common Lisp implementation (2020-)

I contribute to the Steel Bank Common Lisp[1] implementation. Of note are improved *basic block positioning* released in SBCL 2.2.1 which speeds up some code by 70%, and the parallel mark-region garbage collector released in SBCL 2.3.8 which reduces memory requirements and reclaims memory faster.

I cleaned up the *hash tables* module and then wrote much of the *compiler backend* for the SICL Common Lisp implementation. I refactored and documented the hash tables in May 2020, ported Abseil's `flat_hash_map` to Common Lisp in February 2021, and implemented *register allocation* and lowering IR to assembly for *x86-64* from April to October 2021.

Regular expression compilation (2020-)

one-more-re-nightmare compiles regular expressions to Common Lisp code for fast text matching.[2] I wrote optimisation passes which generate simpler automata, and accelerated searching of prefix literals by generating *single instruction-multiple data* code.

Object-capability programming system (2022-)

I work on the Utena interactive object-capability programming system, by designing the *abstract machine* and semantics of the system,[3] and implementing the system in Common Lisp (for practical use) and Standard ML (for specifying semantics).

Replicated object system (2018-2021)

I designed and implemented the *replicated object system* Netfarm from 2018 to 2021 for creating portable and modular decentralised applications. It includes a Kademlia-based *distributed hash table*, work distribution for efficient retrieval of new objects using fine-grained locking which I modelled in *TLA*$^+$, and a content moderation scheme involving *collaborative filtering*.

PUBLICATIONS

Hayley Patton. Parallel garbage collection for SBCL. In *Proceedings of the 16th European Lisp Symposium (ELS'23)*, 2023. https://zenodo.org/record/7816398

Hayley Patton. A replicated object system. In *Proceedings of the 14th European Lisp Symposium (ELS'21)*, 2021. https://zenodo.org/record/4712699

EDUCATION

Attended Bachelor of Computer Science at La Trobe University in 2020. Studied *C++* and *Java*.

Achieved Bachelor of Computer Science with Distinction at RMIT University. Studied *C++*, *Erlang*, *Java*, *Python*, and *SQL* (with *MySQL* and *Postgres*). My capstone project was exhaustive search of a theoretical game, using SIMD for fast equivalence detection and SIMD and *OpenCL* for the remaining search.

SKILLS

Common Lisp, *C* and *Java* programmer. Also familiar with *Erlang*, *Python*, and *Standard ML*. Infrequent designer of parallel and concurrent algorithms. Works with *Linux* and *Git*. Typesets documents with LaTeX.

[1] https://sbcl.org/

[2] https://github.com/telekons/one-more-re-nightmare

[3] https://cal-coop.gitlab.io/utena/utena-specification/main.pdf