What's New in the Community Books

Since the ACL2-2022 Workshop

Alessandro Coglio[1,5,6] (presenter),
Warren Hunt[3,8], Matt Kaufmann[3],
Ankit Kumar[7], Vivek Ramanathan[3],
David Russinoff[2], Eric W. Smith[5,6,1],
Sol Swords[4], Max von Hippel[7]

[1]Aleo Systems Inc., [2]Arm Inc., [3]ForrestHunt Inc.,
[4]Intel Corp., [5]Kestrel Institute, [6]Kestrel Technology LLC,
[7]Northeastern University, [8]University of Texas at Austin

ACL2-2023 Workshop

Overview

- Almost 6,000 non-merge commits since the last Workshop.
- From several contributors from several organizations.
- Spanning hardware, mathematics, cryptography, blockchain, programming languages, virtual machines, machine code, standards, analysis, synthesis, and more.
- These slides provide succinct summaries of major items in the book release notes, ordered by book path within each of the new and improved library parts.

New Libraries

kestrel/acl2data: Tools to collect machine learning data from the community books.

- Collected data is put into `*__acl2data.out` files.
- Data is collected by breaking proofs systematically and recording resulting information, including checkpoints.
- More information in `README` and `gather/README`.
- Used by Proof Helpers in `kestrel/helpers`.

New Libraries

`kestrel/big-data`: Analyzing the ACL2 books as an object of scientific study.

- Work on identifying a maximal set of non-conflicting package definitions.

New Libraries

`kestrel/built-ins`: Tools to collect, document, and disable built-in events.

- Collect all the built-in event names in constants for `defuns`, `defthms`, `defaxioms`, etc.
- Define XDOC topics for different portions of those events (by rule class, by data types, etc.).
- Provide macros to disable certain sets of built-in events, e.g. all logic-mode functions, or all 'default' rules.

New Libraries

`kestrel/hints`: Utilities for manipulating hints and parts of hints (combining, removing, renaming).

- Collected and improved existing utilities, and added new ones.
- New `:casesx` hint to make all 2^n combinations of n boolean expressions.

New Libraries

`kestrel/tests`: Tests of various constructs, to understand precisely what they do in all cases.

- Add tests of `b*`, `case-match`, `mv-let`, `defthm`, `defrule`, `xargs`, and `hints`.

New Libraries

`kestrel/world-light`: Lightweight world utilities.

- Querying the world, gathering event names, etc.

New Libraries

`kestrel/zip`: Handling zip files.

- Can unzip a file within ACL2.
- Only handles Deflate compression (or no compression).

New Libraries

`projects/vwsim`: Super-conducting circuit simulator system.

- Presented at the previous Workshop, now available for general use.
- Simulator of RSFQ (Rapid Single Flux Quantum), a digital electronic device that uses superconducting devices to process digital signals.

New Libraries

`workshops/2023/kumar-etal`: Formal model of GossipSub protocol with various attacks. (Discussed in a paper at this Workshop.)

- Model of entire protocol with special attention paid to score function.
- Proof that the protocol satisfies some security properties but not others.
- Examples of automated attack discovery using CGen.

New Libraries

workshops/2023/vonhippel-etal: Formal model of Retransmission TimeOut (RTO) computation. (Discussed in a paper at this Workshop.)

- Proofs in ACL2, ACL2s, and ACL2(r) that for all $\alpha \in [0, 1)$, $\lim_{n \to \infty} \alpha^n = 0$.
- Multiple proof strategies in ACL2s.
- Formal model of RTO system and proven asymptotic bounds on its variables.

Improved Libraries

build: `cert.pl` build system.

- New support for builds that involve saved images.
- Support for `ACL2_PROJECTS` file specifying project directories for book relocation.
- `build/include-events`: utility for including books that are bundles of other books without needing to certify the bundles.
- For Jenkins: improve `Makefile`, improve script printing and documentation, and limit the max load.

Improved Libraries

centaur/sv and **centaur/vl**: Centaur/Intel hardware verif framework.

- New methodology for datapath proofs with cutpoint decomposition.
- New dual-run comparison mode for SVTV-CHASE hardware simulation debugging tool.
- Many bugfixes and small improvements.

centaur/fgl: Centaur/Intel bitblasting rewriter.

- New capability to solve conjectures with 37 or fewer Boolean variables by exhaustive testing.
- Many small improvements and added features.

Improved Libraries

demos: Demos of ACL2 and tools built on ACL2.

- Added examples of `include-raw`.
- Added examples of `partial-encapsulate`.

Improved Libraries

doc: ACL2 + Books Documentation.

- Added Top 100 Theorems list for ACL2.

Improved Libraries

`kestrel/abnf`: Augmented Backus-Naur Form (ABNF).

- Refactored to organize constituents more clearly.
- Significantly extended the parsing generation tools.
- Added new tools for ingesting grammars and generating operations and theorems for them.

Improved Libraries

`kestrel/acl2-arrays`: Reasoning about ACL2 arrays (`aref1`, `aset1`, etc.).

- Added and improved rules.
- Improved organization and documentation.

Improved Libraries

`kestrel/alists-light`: Lightwight library about alists.

- Added and improved rules.
- Added `string-string-alistp`, `clear-keys`.
- New book on `symbol-symbol-alistp`.
- Deprecated `lookup-equal-lst` (use `map-lookup-equal`).

Improved Libraries

`kestrel/arithmetic-light`: Lightwight library about arithmetic.

- Added and improved rules.
- Added `log2` (floor of log, for any positive rational).

Improved Libraries

`kestrel/arrays-2d`: Two-dimensional arrays as lists of lists.

- Various improvements (some suggested by `improve-book` tool).

Improved Libraries

`kestrel/apt`: Automated Program Transformations (APT).

- Improved the `simplify`, `wrap-output`, `drop-irrelevant-params`, `rename-params`, and `finite-difference` transformations.
- Improved the robustness of the proofs generated by the `restrict` transformation.
- Added the `def` utility for calling transformations.
- New transformation `rename-calls`.

Improved Libraries

kestrel/axe (1): The Axe Toolkit.

- Many improvements and fixes (clarified code, added and verified guards, removed skip-proofs from legacy tools, new proofs about implementation, deprecated old code, fixed names, more robust proofs).
- Added, improved, and organized rules.
- Improved rule-lists and built in more rules.
- Improved printing, error handling, and reporting.
- Improved counterexample machinery.
- Optimized: Use hash-tables and fast-alists, avoid some quadratic behaviors, use `tshell-call` instead of `sys-call`.

Improved Libraries

kestrel/axe (2): The Axe Toolkit.

- Improved pruning (apply to DAGs, use more assumptions, evaluate new ground terms) and use during lifting.
- Evaluate more ground terms. Improve use of memoization.
- The ACL2_STP_VARIETY environment variable now governs how STP is to be called (which option syntax).
- Proved some theorems validating Axe's translation to SMT solvers.
- Improved `unroll-spec-basic` (set of rules used).
- Added tests. Improved documentation.

Improved Libraries

`kestrel/axe (3)`: The Axe Toolkit.

- Added `:max-conflicts` option to query tool.
- Renamed `prove-equivalence2` to `prove-equal-with-tactics`.
- Allow each generated prover to have its own set of `default-global-rules`, add `:extra-global-rules` option.
- Fix rare completeness issue in provers.
- Add `:var-ordering` option to prevent undesirable substitutions.
- Add new prover tactic `:rewrite-top`.

Improved Libraries

`kestrel/axe/jvm`: The Axe Toolkit for JVM.

- Added JVM Formal Unit Tester.
- Simplified `unroll-java-code` in several ways.
- Added examples: Axe proofs of AES implementations, Formal Unit Tester examples.

`kestrel/axe/r1cs`: The Axe Toolkit for R1CSes

- Built in more global rules.

Improved Libraries

`kestrel/axe/x86`: The Axe Toolkit for x86.

- Added prototype Formal Unit Tester for x86 binaries.
- Added support for parsing and lifting ELF files.
- Misc improvements to x86 lifting.
- More readable normal forms, better debugging.
- Added `:position-independent` option.
- Added examples: TEA block cipher, recursive factorial, popcount.

Improved Libraries

`kestrel/booleans`: Rules and definitions involving booleans (supports Axe).

- New and improved rules.

Improved Libraries

`kestrel/bv`: Bit vectors.

- Added, improved, renamed, and organized rules. Improved names.
- Macro fixes. Tweaked `repeatbit`.
- New scheme for converting terms to use bv functions.
- Added `bitxor$`, `bitand$`, and `bitor$` (faster, stricter guards).

Improved Libraries

`kestrel/bv-lists`: Lists of bit vectors.

- Added and organized rules.
- New bv-array conversion utilities.

Improved Libraries

`kestrel/c`: Models, proofs, and tools for C.

- Refactored to organize constituents (deep embedding, shallow embedding, code generator) more clearly.
- Extensions and improvements to the model of C.
- Significant extensions of the C code generation capabilities.
- Significant performance improvements of the generated correctness proofs, via a new modular proof approach.

Improved Libraries

`kestrel/clause-processors`: Clause processors and utilities for them.

- Organize library, misc cleanups.

Improved Libraries

`kestrel/crypto`: Cryptography.

- Added a formal spec for the AES block cipher.
- Added a formal spec of the SHA-3 hash function (proved equivalence to Keccak spec in some cases).
- Added a formal spec of the TEA block cipher.

Improved Libraries

`kestrel/crypto/pfcs`: Prime Field Constraint Systems (PFCS). (Discussed in a paper at this Workshop.)

- Added a concrete syntax.
- Improved the abstract syntax.
- Added many theorems, including proof support rules for compositional reasoning.
- Added a translator from R1CS to PFCS, along with a checker for the R1CS subset of PFCS.
- Added a proof-generating lifter from deeply to shallowly embedded PFCS.

Improved Libraries

`kestrel/crypto/r1cs`: Rank-1 constraint systems.

- Added and generalized rules.
- Added dense form of R1CSes (legacy).
- Added optimized function `r1cs-constraint-list-vars`.
- Imported `primep` into package.

Improved Libraries

`kestrel/evaluators`: Evaluators (e.g., to support meta-reasoning).

- New tool `defevaluator-theorems`.
- Added theorems.

Improved Libraries

kestrel/file-io-light: Lightwight library about file input and output.

- Added rules about `princ$`, `prin1$`, `open-output-channel`, `open-output-channel!`, etc.
- New books on `close-input-channel`, `close-output-channel`, `typed-io-listp`, `print-object$-fn`, and `newline`.
- Added `read-objects-from-file-with-pkg`, `read-file-into-line-list`, and `read-file-into-line-list-no-error`.
- Added and improved rules. Organized and optimized. Fixed names.
- Proved that various functions don't change the world.

Improved Libraries

`kestrel/floats`: Cleanroom formalization of IEEE-754 floating point (in progress).

- Model of floating point formats, representable rationals, encoding/decoding, normals, subnormals, zeros, infinities, NaNs, comparisons. No rounding yet!
- Tries to match the official spec as closely as possible.
- Proved some connection theorems to the RTL formalization.

Improved Libraries

`kestrel/helpers`: Helper tools for finding / repairing proofs and improving ACL2 books.

- New Proof Advice tool to get proof help over the web, based on ML models and other heuristics.
- Tools to evaluate proof advice on one or many books.
- Improvements to `improve-book` tool. Improve linter and move it here.
- New `speed-up-event` utility that tries to speed up theorems.
- Add initial `repair` tool to fix broken proofs.

Improved Libraries

`kestrel/htclient`: Communication via HTTP.

- Lightweight variant of HTTP POST.

Improved Libraries

`kestrel/json-parser`: JSON Parser.

- Added `parse-string-as-json`, `defconst-from-json-file`.
- Improved implementation (more tail recursion).
- Added and improved rules.
- Organized and sped up proofs.
- Added and improved tests.

Improved Libraries

`kestrel/jvm`: JVM model.

- Clarified model and added new rules.
- Better normal forms and abstractions.
- Renamed `load-class-XXX` to `read-class-XXX`.
- Support for reading code directly from jar files.
- Put more names in `JVM` package.
- Worked on verifying guards.
- Improved handling of floats/doubles, LDC bytecode.
- Improved error reporting in class file parser.

Improved Libraries

`kestrel/lists-light`: Lightweight lists library.

- Added various rules, organized library.
- New books on `position-equal-ac`, `position-equal`.
- Added `filter-non-members`.
- Added `union-equal-alt` (keeps earlier duplicates).

Improved Libraries

`kestrel/meta`: Metafunctions / meta rules.

- Add some experiments verifying rewriter-like tools using `meta-extract`.

Improved Libraries

`kestrel/number-theory`: Number theory library.

- Import some symbols into the `PRIMES` package, including `dm::primep`.

Improved Libraries

`kestrel/prime-fields`: Prime field library.

- Add and improve rules, and simplify.
- Import some symbols into the `PRIMES` package, including `dm::primep`.

Improved Libraries

kestrel/std/system: Standard system library (extension of Std).

- Added utilities `untranslate$`, `genvar$`, `one-way-unify$`, and `termination-theorem$`, which are logic-mode variants of built-in utilities, via `magic-ev-fncall`.
- Added utilities `guard-theorem-no-simplify` (program-mode) and `guard-theorem-no-simplify$` (logic-mode), which are variants of `guard-theorem` that does no simplification.

Improved Libraries

`kestrel/std/util`: Standard system library (extension of Std).

- New utilities for error-value tuples have been added, to facilitate the generation, propagation, and catching of errors in statically strongly typed code.

Improved Libraries

kestrel/strings-light: Lightwight library about strings.

- Added `string-starts-withp`, `add-prefix-to-strings`, `split-string-last`, `strip-prefix-from-string`, `strip-suffix-from-string`, `strip-suffix-from-strings`, `decimal-digit-to-char`, `decimal-digit-to-string`, `strnthcdr`.
- New books about subseq, `strcar`, `strcdr`, `char`.
- Renamed `string-ends-inp` to `string-ends-withp`
- Added rules and tests.

Improved Libraries

`kestrel/terms-light`: Lightweight library for manipulating terms.

- New and improved rules. Various fixes. Generalized, optimized, and clarified.
- New books about `all-fnnames1`, `trivial-formals`, `get-conjuncts`, and `get-hyps-and-conc`, and `replace-corresponding-arg`.
- Improved `reconstruct-lets-in-term` to handle `mv-let` and ignored vals.
- Improved `substitute-unnecessary-lambda-vars`.
- Added `subst-var-alt` (avoids creating unserialized lambdas).
- Proved that some utilities return / preserve closed lambdas.
- Added tests.

Improved Libraries

kestrel/typed-lists-light: Lightweight library for lists of elements of various types.

- Added rules and organized library.
- New books about `alist-listp`, `symbol-alist-listp`, `symbol-term-alist-listp`, `map-strip-cdrs`, `all-digit-charsp`.
- Added `string-list-listp`.

Improved Libraries

`kestrel/untranslated-terms`: Utilities about untranslated terms.

- Various improvements, especially about case, `case-match`, and cond.
- Improvements for b* (e.g., support for &, `list`, and `er` binders).
- Added utilities for extracting conjuncts and disjuncts, and for getting free variables from terms.
- Improved support for `let` and `mv-let`.

`kestrel/untranslated-terms-old`: Old utilities about untranslated terms.

- Moved to `kestrel/untranslated-terms/untranslated-terms-old`.
- Fixed some checks.

Improved Libraries

`kestrel/utilities (1)`: Various utilities in the Kestrel books.

- Added `show-books` utility, which returns a tree representing the books included in the current ACL2 session.
- Improved `checkpoint-list` and related utilities.
- Added parser for CSV files and utils for handling parsed CSV files.
- Split `byte-array-stobj` into separate book.
- Proved that `merge-sort-symbol<` returns a sorted list.
- Deprecated `defopeners-mut-rec` (use `defopeners`).

Improved Libraries

kestrel/utilities (2): Various utilities in the Kestrel books.

- New books on read-acl2-oracle, update-acl2-oracle, invariant-risk, getenv$, get-serialize-character, plist-worldp, char-code, widening margins, strip-cadrs, >=-len, our-digit-char-p, read-run-time, get-cpu-time.
- Added misc rules (e.g., about world, state, channels, assoc-keyword, setting margins, coerce, non-trivial-bindings, acl2-count).
- New utilities: most-recent-failed-theorem-goal, translatable-term-listp, parsing of strings, dir-of-path, all-included-books, defthms-in-world, sys-call-event, split-path, reduce-print-level, pack-in-package, eval-tests, fast-alist-set, array-stobj, setenv$-event, prove$-nice-trying-hints, prove$-nice-with-time-and-steps, last-prover-steps$, merge-sort-string<.

Improved Libraries

kestrel/utilities (3): Various utilities in the Kestrel books.

- Improved `prove$-nice` and `prove$+`, `declare` utilities, `defmergsort`, `defxdoc-for-macro`, utilities for temp dirs, and material about conjuncts/disjuncts and `acl2-count`.
- New tests. Better error reporting.
- Improved utilities: `defmergesort` (`:extra-theorems` option), `ld-history` utils, `ubi`, `defstobj+` (generate more theorems, e.g., about hash-table fields).
- New utils `shuffle-array-stobj` / `shuffle-array-stobj2` and `shuffle-list` / `shuffle-list2`, using a Fisher-Yates shuffle.

Improved Libraries

`kestrel/utilities/digits-any-base`: Number representations in arbitrary bases.

- Added several theorems.
- Extended the `defdigits` macro to generate more theorems.

Improved Libraries

`kestrel/utilities/omaps`: Ordered maps (omaps).

- Added several theorems.
- Improved documentation and organization.

Improved Libraries

`kestrel/x86`: Kestrel's additions to support working with the `x86isa` model

- Improved and added various rules, e.g., about condition codes.
- Clarified and improved rule lists (used by Axe).
- Improved parsers, fixed ELF parser bug.

Improved Libraries

`kestrel/zcash`: The Zcash protocol.

- Add `:var-ordering` option to `verify-zcash-r1cs`.

Improved Libraries

`projects/curve25519` and **`projects/shnf`**: Older developments by Russinoff.

- Switched to DM (discrete math) package.

Improved Libraries

`projects/group` and **`projects/numbers`**: Formalizations of group theory and number theory. (Discussed in two papers and a rump talk at this workshop).

- Elementary finite group theory, essentially the content of an advanced undergraduate course.
- Various results in elementary number theory.
- `projects/numbers` was previously called `projects/quadratic-reciprocity`, and it has been moved to the DM package.

Improved Libraries

`rtl`: Register Transfer Logic.

- Switch some utilities, such as `local-defun`, to ACL2 package.

Improved Libraries

`std/io`: Standard I/O library.

- Extended `read-string` to take an optional package as argument.
- Added a lightweight version of `read-string`.

Improved Libraries

`std/lists`: Standard Lists library.

- Have `deflist` add `-compound-recognizer` to some rule names, to avoid clashes.

Improved Libraries

`std/util`: Standard utilities library.

- The `er` binder of `b*` has been extended with an option `:iferr` to return an alternative value in case of error.
- Calls of `defaggregate` can now (usually) be local, and its proofs are more robust.
- Calls of `define` no longer turn on error output if it was off before the `define`.

Improved Libraries

`tools`: Various tools.

- Substantially enhanced `with-supporters`.
- Improved `prove$`, e.g. to treat hard errors as ordinary failures.
- Made `run-script` more robust against time variations.
- Added tools `functions-after` and `macros-after`, which return names of functions and macros introduced after a given name.
- Added new `tools/top` book to gather tools (for inclusion in the manual, instead of individual tool books).

Improved Libraries

`workshops/2022/russinoff-calendar`: Modeling the Hebrew calendar.

- Switched to ACL2 package.

`workshops/2022/russinoff-groups`: Group theory.

- Switched to DM (discrete math) package.

Improved Libraries

xdoc: XDOC documentation.

- Added support for Greek letters and more mathematical symbols.