LKB-FOS Update

John Carroll
Department of Informatics, University of Sussex, UK

DELPH-IN Summit, July 2021

Outline

Functionality: releases, architectures, spanning-only rules

User interface: menus, tree windows, known issues

Scalability: improvements, comparisons, parse time breakdown

Internals: bug fixes, global state reduced

Functionality

- 3 LKB-FOS releases since last year's summit
  - 27 Dec 2020 (bug fixes)
  - 30 Apr 2021 (major)
  - 18 Jul 2021 (minor)

- Latest release includes a native executable for Apple M1
  - x86 binary can be run in emulation, but occasionally triggers exception *'GPR thread_set_state is unsupported while in sa_tramp'* (c.f. .NET bug)

- Added the ability to declare rules as 'spanning-only'; when parsing, such rules will only ever be applied over the entire input string. E.g.

```
(defparameter *spanning-only-rules* '(aj-hd_int-inv_c aj-r_frg_c))
```

User Interface

- Menus
  - modern menu navigation: click to open, click again to select a menu item
  - LKB Top commands 'Options > Shrink/Expand menu' work more smoothly
- Parse tree windows
  - trees balanced
  - window opening and resizing more responsive with 100s of parses
  - bug fix: node labelling takes account of FS re-entrancies at top level
- Known issues due to McCLIM bugs (fixes expected)
  - large feature structures can become garbled after scrolling
  - LKB Top sometimes omits blank lines

Scalability

- Reduction in garbage collection (GC) overheads in batch processing
  - with modern GCs, GC time is proportional to amount of *live* data in the memory area being GCed[1]
  - between sentences, little live data in 'nursery' if temporary dags removed
  - so only GC between sentences, and only if >2GB allocated since last GC

  → GC time reduced from 20% to 2% of total CPU time

- More economical hyperactive parsing strategy
  - count number of times each active edge gets reconstructed
  - if greater than a (small) threshold then permanently copy the edge's dag
  - fewer unifications replayed; more copies, but much more structure shared

  → 15% faster, 20% less memory

[1] N.B. GC time is not proportional to the amount of *allocated* data. For technical background see https://medium.com/@MartinCracauer/generational-garbage-collection-write-barriers-write-protection-and-userfaultfd-2-8b0e796b8f7f

- Tighter packing in parser
  - if each rule has a different top type then fewer opportunities to pack nodes
  - generalise top type of FS on non-unary passive edges

  $\rightarrow$ 8% fewer edges

- More efficient quickcheck
  - each quickcheck executes up to 30 or so type unifications, each involving at least a hash table lookup
  - replace type unification with a single logical *and* applied to bit encodings of possible leaf types – in cases where there's no more than 64 of these

  $\rightarrow$ 5% faster

- More economical calling of the unifier
  - in a hang-over from YADU(?), each attempt to unify a rule with a FS prepended fresh dag structure representing the path to the daughter node
  - this is unnecessary with (quasi-)destructive unification and no defaults

  $\rightarrow$ 10% faster, 10% less memory

- More appropriate data structures
  - active edge chart in parser
  - generator chart
  - rule application filter

Comparisons

Parsing Rondane (1424 items), ERG 2018, compute top-ranked parse

Start-symbols `root_xxx` × 5, resource limits giving ~25 timeouts, spanning-only rules, packing, unpacking top-ranked parse. Total numbers of packed parses comparable. Running on Intel iMac 3.3GHz i5 16GB, M1 MacBook Pro 3.2GHz 16GB

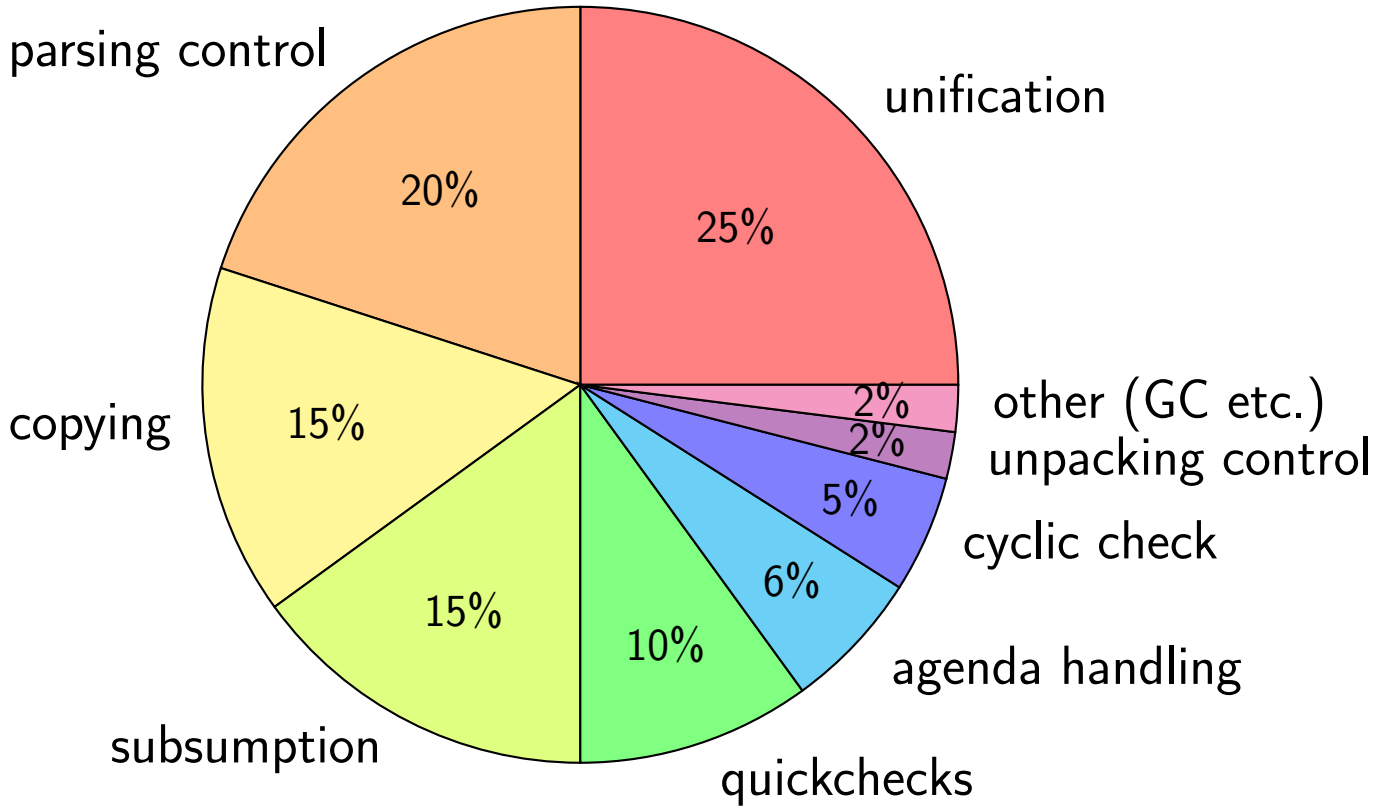
	CPU (mm:ss)
Intel	
ACE *	15:24
ACE (with `--disable-generalization`) **	*28:05*
LKB-FOS ***	27:51
M1	
ACE emulated *	10:54
LKB-FOS emulated ***	*16:46*
LKB-FOS native ***	15:24

* version 0.9.34-m1, successfully parsed 1354 items, options `-1 -R`

** successfully parsed 1355 items, options `-1 -R --disable-generalization --max-chart-megabytes=2000 --max-unpack-megabytes=2500`

*** release 18 Jul 2021, successfully parsed 1149 items, parameters `*first-only-p* 1, *maximum-number-of-tasks* 180000, *maximum-number-of-edges* 200000, *unpack-edge-allowance* 50000`

Parse time breakdown



Type unification 10%, split between quickchecks, unification, subsumption

Internals

- [incr tsdb()]: profiles containing integers longer than 30 bits; symbolic links
- LUI interface: interactive unification type constraint failures; types with no features
- View... command dialog: names starting with or consisting only of digits
- Reading transfer and MRS rules: revised TDL syntax specification
- Generator: default value of `mrs::*normalize-predicates-p*` assumes a SEM-I; `*gen-extract-surface-hook*` need not be set; fixes for `*gen-start-symbol*` and `*additional-root-condition*`; `extract-pred-from-rel-fs` changed to LOGON version
- Parser and generator: `*non-idiom-root*` checked before invoking `*additional-root-condition*`
- Removed/disabled some global state: chart edge 'registry', dag pool

Summary

Continuing development over the past year

- native binary for new M1 Macs
- improvements to user interface and processing speed
- bug fixes

Still to do

- chart mapping
- selective unpacking grandparenting
- unified grammar configuration file format
- Windows version
- Homebrew installation