LKB-FOS Update

John Carroll

University of Sussex, UK

DELPH-IN Summit, July 2024

Outline

One LKB-FOS release since last year's summit, on 28 June

   Bug fixes

   Improvements: smoother interface, some core code cleaned and modernised

   New experimental feature: parser local ambiguity packing under generalisation

To-do list

US University of Sussex

Bug Fixes

Fixed bugs and missing functionality in DELPH-IN YY input mode

- allow a token to have multiple inflections

- apply pre-processor inflections to generic lexical entries

- correctly interpret the inflection 'null'

Students on Ling 567 reported a bug that prevented generation of sentences containing multiple occurrences of semantically empty lexical items

- prompted discussion on DELPH-IN Discourse about breadth-first search and outputting strings from an incomplete generation forest

In [incr tsdb()], attempting to treebank sometimes led to error `The value NIL is not of type NUMBER`

US University of Sussex

Improvements

Smoother interface

- more consistent and informative diagnostic and progress textual messages; corrected documentation strings for some LKB parameters

- graphical interface a bit more polished

  - faster and artefact-free window display and scrolling

  - able to use mouse to move cursor and select text in dialog boxes

- better support for high DPI displays – add `(setq mcclim-truetype::*dpi* 96)` to your ~/.lkbrc

University
of Sussex

Some core code cleaned and modernised, especially in core unification functions, type unification, agenda handling

Examples:

1. agenda implemented as a priority queue

   - algorithm reference books present queue updates as a sequence of element swaps – but can be done better
   - priorities of new elements coerced to single floats $\rightarrow$ much faster type checking and execution of priority comparisons

2. Low-level DAG slot access guarded by a 'generation' counter

   - encourage compiler to emit machine code that checks the counter using branchless conditionals

Code clean-up gives a 10% improvement in parse time

University
of Sussex

Local Ambiguity Packing

The parser can now pack local ambiguity under feature structure generalisation.
Inspired by a similar facility in ACE. Enable by `(setq *generalising-p* t)`

```
procedure dag-subsumes-p(dag1, dag2) ≡
    (forwardp, backwardp) ←                                    {establish context for non-local exit}
        catch with tag 'fail' dag-subsumes-p0(dag1, dag2, true, true);
    invalidate-temporary-pointers();                           {reset temporary 'copy' pointers}
    return (forwardp, backwardp);
end

procedure dag-subsumes-p0(dag1, dag2, forwardp, backwardp) ≡
    if (dag1.copy is empty) then dag1.copy ← dag2;             {check reentrancies}
    else if(dag1.copy ≠ dag2) then forwardp ← false; fi
    if (dag2.copy is empty) then dag2.copy ← dag1;
    else if (dag2.copy ≠ dag1) then backwardp ← false; fi
    if (forwardp = false and backwardp = false) then
        throw (false, false) with tag 'fail';                 {reentrancy check failed}
    fi
    if (not supertype-or-equal-p(dag1.type, dag2.type)) then forwardp ← false; fi    {check types}
    if (not supertype-or-equal-p(dag2.type, dag1.type)) then backwardp ← false; fi
    if (forwardp = false and backwardp = false) then
        throw (false, false) with tag 'fail';                 {no subtype relations}
    fi
    for each arc in intersect(dag1.arcs, dag2.arcs) do         {check shared arcs recursively}
        (forwardp, backwardp) ←
            dag-subsumes-p0(destination of arc for dag1, destination of arc for dag2, forwardp, backwardp);
    od
    return (forwardp, backwardp);                              {signal result to caller}
end
```

re-entrancies

types

follow arcs

from Oepen & Carroll (2000)

Still experimental; only tested thoroughly on the ERG and partially on the SRG

US University of Sussex

Practical results

Parsing Rondane with stable 2023 version of ERG, no PoS tagging, computing top-ranked parse, resource limits giving ~25 timeouts

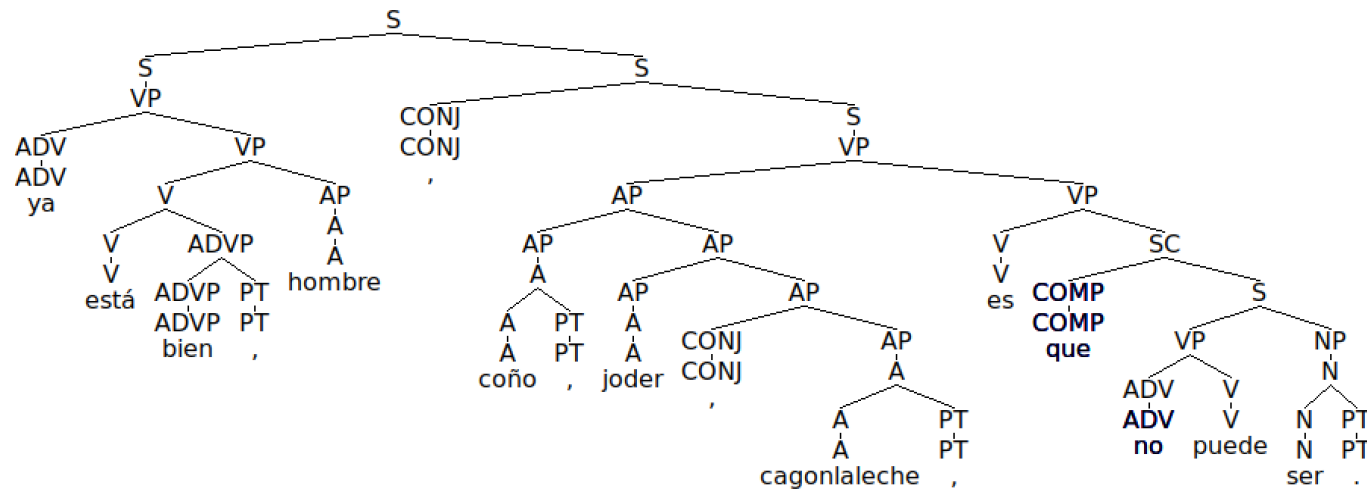| *Mac Intel* *	*mm:ss*	*Mac M1* †	*mm:ss*
LKB-FOS	10:17	LKB-FOS native	7:26
ACE	13:54		

* iMac i7 3.8GHz

† MacBook Pro M1

University
of Sussex

Parsing with SRG version of February 2024

First 20 items of 'an12' test suite parse in 7.7 sec (LKB-FOS), 72 sec (ACE)

On item 21, ACE fails to terminate, even with a timeout; LKB-FOS parses it successfully in 29 sec

University of Sussex

Summary

Development has continued over the past year

- bug fixes

- improvements

- new features

Still a long to-do list, including

- change post-generation chart mapping to act on 'full' FS, not 'edge' FS

- remove passive chart parser

- add 'grandparent' features in selective unpacking

- unified grammar configuration file format

- part of speech tagger

- Microsoft Windows version

University of Sussex