The Grammar Codex and the Linguistic Type Database

Francis **Bond**

Department of Asian Studies,
Palacký University, Olomouc, Czechia

<bond@ieee.org>

DELPH-IN 2025

Faculty of Arts

Introducing the Grammar Codex

- DELPH-IN has produced many grammars, but they are not always easy to find
- So let's make them (i) findable in one place, (ii) accessible with latest tools, (iii) documented
  - Compiled a list of grammars
  - Download, patch were necessary (push upstream if we can)
  - Parse with ltdb (pydelphin)
  - Parse with ace

Faculty of Arts

How does it work

- Has a toml file with the urls (human readable)
- Have a local directory with changes, just over-write
- After we download a grammar use the METADATA file
  - Now parse it as toml
  - This works for current files, but allows multi-line entries
  - Should work for Grammarium (Luis's demo)
- We must have at least three fields
  - GRAMMAR_NAME
  - SHORT_GRAMMAR_NAME
  - ACE_CONFIG_FILE (path relative to METADATA)
  - TSDB_ROOTS (optional, defaults to `gold`)
  - PROFILES (optional, defaults to ALL)
- Code available here:
  - https://github.com/delph-in/codex
  - https://github.com/fcbond/ltdb

Faculty of Arts

Queries, Requests and Dreams

- Would anyone object to me uploading the complete collection to github as a release (and to Zenodo)
- Is everyone ok with making the metadata toml
- If you have or know of a grammar that is not in the codex and should be, let me know!
  - Can we/should we add from Emily's class (567)?
  - It would be nice to have the book grammars now in LKB, maybe split?
- It would be good to also have everything run with the LKB
- Please try to run it, and let me know if it works for you
- Would like to integrate with Luis's demo
- Would like to host at UPOL and VU (and maybe elsewhere)
- Would like to add grew-match (UD tree search)
- Can we have `version` in an easier to parse form?

Faculty of Arts

A historical view

- Open Multilingual Wordnet for Grammars!
  - Make it easier to look up and compare
  - DELPH-IN grammars better formatted!
- LOGON in python
  - Bringing bits together rather than bundling it all up
  - It would be good to support LISP too

Faculty of Arts

Why the LTDB?

- It is hard to work on a grammar that you did not write (just like any software)
  - Or that you wrote in collaboration
  - Or that you wrote sometime ago
  - Or that is generated by the MATRIX
- It is hard to be consistent within a treebank
  - Especially if it has multiple annotators
  - Or that you treebanked some bits some time ago
  - Or just if it is very big
- LTDB is an attempt to store the information you had in your mind when you wrote the grammar and make it more accessible
  - inspired by literate programming (Knuth, 1992)
  - store documentation **about the grammar** — **in the grammar files**

Faculty of Arts

LTDB

- Completely rewritten Lexical Type Database (Hashimoto et al., 2007a,b)
- Generalized in 2014 to handle all types (and some instances)

The Linguistic Type Database

status	thing	source	endi
type	normal type		
ltype	lexical type	(type and in lexicon)	lt
rule	grammar rule	(LKB::*RULES)	c
lrule	lexical rule	(LKB::*LRULES)	
irule	inflectional rule	(LKB::*LRULES and (inflectional-rule-p id))	
root	start symbol	(LKB::*root-entries*)	

Rules also list number of daughters and head daughter.

Faculty of Arts

Headedness

We are **Head-driven** Phrase Structure Grammar, so it is nice to know the headedness of rules. We record 5 different possibilities:

- ▲ unary: headed
- △ unary: non-headed
- ◭ binary: left-headed
- ◮ binary: right-headed
- ◬ binary: non-headed

For each rule, in look for the daughters of the rule, see if `*head-daughter-path*` exists (only implemented for LKB at the moment).
Now read from `rules.hds` **NEW**

Faculty of Arts

Uses the TDL doc string

```
n_-_c_le := n_intr_lex_entry
"""Intransitive count noun (icn)
<ex>The dog barked.
<nex>Much dog bark.""".
```

Originally:

```
; <type val="n_-_c_le">
; <description>Intransitive count noun (icn)
; <ex>The dog barked.
; <nex>
; <todo>
; </type>
n_-_c_le := n_intr_lex_entry.
```

Faculty of Arts

Other Changes

- Integration with grammar catalogue
- Description written in Restructured Text
  - Allows more flexible formatting
  - Special macros for positive and negative examples
- Scripts written in python3
- Source available in github:
  https://github.com/fcbond/ltdb

Faculty of Arts

2020 enhancements

- ACE, LKB and PET now allow docstrings with """ """ on all types and instances, to read them all
  - Thanks everyone for their support.
- The fftb can link to this for rules and lexical types
  - Maybe we should include an LTDB url in the metadata
- Moved to python3
- Now read tdl with PyDelphin
- You can specify a particular grammar (script file or ace config)
  latest version a branch on github, will move next week

Faculty of Arts

2022 Enhancements

- Trees and MRS displayed using javascript (like delphin-viz)
- Search for MRS predicates in the corpus, as well as types and words
- Slightly more robust
- Can read grammars with LKB (using `lkb/script`) or PyDelphin (using `ace/config.tdl`) or both
- Can pre-load some lisp before reading the config file
  e.g. to load the `mal` grammar:

```
./make-ltdb.bash --lisp '(push :mal *features*)'
--script /path/to/grammar/lkb/script
--acecfg /path/to/grammar/ace/config-mal.tdl
```

Faculty of Arts

Other useful information

- Make the conversion logs available (so the grammar developer or user can see if there are any known issues) — typically not all MRS's can be converted to DRMS or JSON
- Give a link to a compressed version of the database, so people can download it — may be easier to access the trees and MRSs for non–delph-in users
  there have been issues with people failing to get MRSs int he past, ...

Faculty of Arts

Major Changes 2023

- Completely rewritten to use **flask** rather than a bunch of cgi scripts
- Each grammar+version is a new DB
  a single ‘grammar’ may have multiple LTDBs
- Only read grammar from ACE
  (LKB was inconsistent with the docstring handling)

Faculty of Arts

2025 Enhancements

- Tested on more grammars, slightly more robust
- Restored look up box
- Now show the examples 20% from the shortest
- Make a profile from the ex/nex/mex
  - parse
  - classify the results `<ex>`
    - ✓ parses with type/rule (in top n)
    - ✗ doesn't parse
    - ? parses without type/rule
  - classify the results `<nex>`
    - ✗ parses with type/rule (in top n)
    - ✓ doesn't parse
    - ? parses without type/rule
- classify the results `<mex>`
  - Same as <ex> but with malrule?

Faculty of Arts

Known ToDos

- Add the Matrix documentation docstrings
- Collate the parse results
  - Add them to the treebank
- Host at Palacký (and maybe alias somehow)
- Look at full tree search over trees and DMRS
  - https://match.grew.fr/
  - WeSearch

Faculty of Arts

Discussion I

- Who is using this? (is it just me and Dan?)
  - How to balance time spent on this and time on a grammar, ...
- Any requests?
- We will try to host ltdb
- It could interface with fftb better
- If we want to annotated features (like INFLECTED of MC), where should this go? In the type that first introduces them? Is there a way to index this (i.e. can we output it automatically from the lkb or pydelphin)?

Faculty of Arts

Discussion II

- We could still chose examples better?
  - Currently I show 8 (could configure)
  - Maybe prefer some profiles over others?
  - Prefer examples in the doc-strings
- I desperately need someone who understands javascript better than me to help a bit (anyone really)
- Should I do MRS/DRMS conversion on the fly?
  People have found the conversion log useful

Faculty of Arts

References I

Chikara Hashimoto, Francis Bond, and Dan Flickinger. 2007a. The lextype DB: A web-based framework for collaborative multilingual grammar and treebank development. In *First International Workshop on Intercultural Collaboration (IWIC-2007)*, pages 44–58.

Chikara Hashimoto, Francis Bond, Takaaki Tanaka, and Melanie Siegel. 2007b. Semi-automatic documentation of an implemented linguistic grammar augmented with a treebank. *Language Resources and Evaluation*, 42(2):117–126. URL http://dx.doi.org/10.1007/s10579-008-9065-9, (Special issue on Asian language technology).

Donald E. Knuth. 1992. *Literate Programming*. CSLI Publications.

Faculty of Arts