TARTU ÜLIKOOL
UNIVERSITAS TARTUENSIS
1632

Type theory and Universal Grammar

Erkki Luuk

CIFMA 2019

Universal Grammar

Introduction

- The idea of Universal Grammar (UG) as the hypothetical linguistic structure shared by all human languages harkens back at least to Roger Bacon in the 13th century [Ranta, 2006]
- The modern notions of UG:
  - Substantive UG [Chomsky, 1970, Chomsky, 1981, Chomsky, 1995]
  - "Diluted" UG: the Language Acquisition Device [Jackendoff, 2002]

Universal Grammar

Introduction

- The ideas of UG occur in the broader context of...
  - Substantive universals [Plank et al., 2009]
  - Implicational universals [Greenberg, 1966]

Greenberg, J. H. (1966). Some universals of grammar with particular reference to the order of meaningful elements. In Greenberg, J. H., editor, Universals of Grammar, pages 73–113. MIT Press, Cambridge, MA.

Plank, F., Mayer, T., Mayorava, T., and Filimonova, E. (2009). The Universals Archive. `https://typo.uni-konstanz.de/archive`

Type theory

Introduction

(0) type := a category of semantic value.

- By (0), type theory is by definition suited for analyzing universal phenomena in natural language (NL), as NL semantics in largely universal (as witnessed by the possibility of translation from any human language to another)

Type theory

Introduction

- Thus, if we could build a fundamentally semantic description of grammar (e.g. one on top of and integrated with a semantically universal description of NL), it might at least stand a chance of being universal

Type theory

Interpreting NL in

- NL expressions as function applications:
  (i) `D man`
  (ii) `run (D (Y man))`
  (iii) `Y love (1st, 2nd)`
  (iv) `man D`
  (v) `m (D an)`
- Complex formulas are written in prefix notation, $a\,b$ or $a(b)$, with a standing for a function and b for argument(s)
- Left-associativity, i.e. left to right valuation

Type theory

Interpreting NL in

- NL expressions as function applications:
  1. `D man`
  2. `run (D (Y man))`
  3. `Y love (1st, 2nd)`
  4. `man D`
  5. `m (D an)`

- Complex formulas are written in prefix notation, $a\,b$ or $a(b)$, with a standing for a function and b for argument(s)

- Left-associativity, i.e. left to right valuation

Specification language

Interpreting NL in

- Call the formulas (i)-(iii) formulas of a specification language (SL)
- Then, we **specify** SL formulas from NL expressions and **derive** NL expressions from SL formulas

Specification language

Axioms

(1) Arguments must be either specified or derived before the relation expressions in which they appear

(2) NL and SL expressions are well-formed both syntactically and semantically, i.e. well-formed and well-typed

(3) For a particular language, the symbols are type constants; in UG they are type variables (e.g. `man` valuates to *man* in English and *homme* in French)

Specification language

in Coq[†]

(vi) `Y know i (who (Y COP ill (the man)))`

(vii) `Y know i (who (Y COP (ill, the man)))`

(viii) `Y know i (who (Y COP (ill, D man)))` : S : $\mathcal{U}$

(ix) `PRES know i (who (PAST COP ill (the man)))`

where S is sentence, $\mathcal{U}$ the top-level universe in SL, and "$x : y$" := "x has type y"

[†] `https://gitlab.com/eluuk/nlc/blob/master/cop.v`

Specification language

in Coq[†]

(vi) `Y know i (who (Y COP ill (the man)))`

(vii) `Y know i (who (Y COP (ill, the man)))`

(viii) `Y know i (who (Y COP (ill, D man)))` $: \mathtt{S} : \mathcal{U}$

(ix) `PRES know i (who (PAST COP ill (the man)))`

where S is sentence, $\mathcal{U}$ the top-level universe in SL, and
"$x : y$" := "x has type y"

[†] `https://gitlab.com/eluuk/nlc/blob/master/cop.v`

Selectional restrictions

Optional

- Selectional restrictions are (onto)logical restrictions on the types of arguments of NL relations (e.g. [Asher, 2014, Luo, 2010])
- For example, an adjective like *red* imposes the restriction that its argument be of type physical entity, while a verb like *know* imposes a restriction that its subject be a sentient and object an informational entity

Selectional restrictions

Optional

(x) $[\mathtt{red}]\colon X_{\mathsf{Phy}} \to \mathtt{P}$

(xi) $[\mathtt{know}]\colon X_{\mathsf{Sen}} \to Y_{\mathsf{Inf}} \to \mathtt{S}$,

where $[x]$ is the interpretation of x, $\mathtt{P}$ is phrase, $\mathtt{S}$ sentence and X, Y are type variables indexed by selectional restrictions

Selectional restrictions

The rule of metaphor or metonymy elimination

$$\frac{u_j : Z_j \qquad (x^e_h : X^e_h) \mapsto (y^e_j : Y^e_j)}{x \ldots (u_j)^e \ldots : W}\ \text{MM-Elim},$$

where x^e_h is a function x, eth argument of which is restricted to h; X^e_h a function type X, eth argument type of which is indexed by h; u_j a (possibly empty) argument u, restricted to j; $x \mapsto y :=$ "x is interpreted as y"; and $X^e_h, Y^e_j, Z_j, W : \mathcal{U}$, where $\mathcal{U}$ the top-level universe in SL

Selectional restrictions

The rule of metaphor or metonymy elimination

- By MM-Elim, whenever we have a metaphor/metonymy (x_h^e is interpreted as y_j^e) and possibly u_j, $x \dots (u_j)^e \dots$ is well typed in SL (and NL)
- For example, $\{\mathrm{idea}_{\mathrm{Inf}}, (\mathrm{red}_{\mathrm{Phy}^1} \mapsto \mathrm{communist}_{\mathrm{Inf}^1})\} \vdash \mathrm{red\ idea}_{\mathrm{Inf}} : W$
- As we take all elementary arguments to be nullary relations, we also have $\{\mathrm{red}_{\mathrm{Phy}^0} \mapsto \mathrm{communist}_{\mathrm{Inf}^0}\} \vdash \mathrm{red}_{\mathrm{Inf}} : W$ for argumental uses of the words

Universals

The problem

- The received view among typological linguists is along the lines that nothing in NL is universal [Haspelmath, 2007, Evans and Levinson, 2009]
- But what about sign, form, meaning, word, sentence, morpheme, phrase, etc.?
- Besides these general counterexamples, the main difficulty is conceptual rather than factual, being due to the virtual non-existence of universally shared definitions

Universals

A proposal: Universal categories

- We propose some universal linguistic categories, defined by their function:
    - proper name (`PN`)
    - connective (`CON` — *and, but, or, not...*)
    - `XP` (Frequently also referred to as NP or DP)
    - declarative sentence (`S` — *john is here...*)
    - interrogative sentence (`IS`)
    - connective composition (`CONC` — *x and y, x or y, not x...*)
- But this is about as far as conventional grammatical categories get us

Universals

Another proposal: Universal supercategories

- We propose some universal linguistic supercategories, defined by their function:
  - case/adposition (`CA` — nominative, accusative, *to*, *from*...)
  - case/adposition phrase (`CAP` — *john*, *him*, *to the house*)
  - numerals/quantifiers (`Q` — *all*, *some*, *no*, *few*, *one*, *two*...)
  - determiner/demonstrative (`D` — *a*, *the*, *this*, *those*...)
  - tense/aspect/mood/voice (`TAM` — the canonical verbal inflection)
  - adverbs or other adverbial phrases (`ADL` — *quickly*, *in a hurry*...)

Flexibles

And such

- To proceed with defining universal supercategories, we use the general polymorphic linguistic category of flexibles [Luuk, 2010], exemplified by words like *sleep* and *run* in English
- Since *sleep* and *run* “flex” between relation and argument, they are flexibles-over-relation-and-argument. Provisionally, we type them `X/R` (with `R` for relation and `X` (from `XP`) for argument)
- There are many categories of flexibles, e.g. *have* is a flexible between auxiliary verb (`AUX`) and infinitival relation (`IR`) (has type `AUX/IR`)

Universals

Core relation and argument

- Now we can posit the universality of the following supercategories:
    - core relation (`R` — verb, copula, infinitival relation, auxiliary verb or flexible-over-relation)
    - core argument (`X` — noun, proper name, pronoun, gerund or flexible-over-argument)
- Examples: an infinitival relation is *like* in *i like to run*, an auxiliary verb is *must* in *i must run* and a gerund is *running* in *running is healthy*

The Coq tests

With a flexible

- Here are some tests of a Coq implementation of the flexible that is polymorphic between function and argument:

```
Check sleep: gs _ _ _ _ _. (*typed as argument*)
Check sleep: NF → _ → S. (*typed as function*)
Fail Check PAST sleep man. (*fails since "man" is not an XP*)
Check PAST sleep (a man). (*a man slept*)
Check PAST sleep (few (PL man)). (*few men slept*)
Check PAST sleep (a (few (PL man))). (*a few men slept*)
Check a sleep.
Fail Check PAST sleep (a hut). (*a hut slept: wrong SR*)
Fail Check PAST sleep (a sleep). (*a sleep slept: wrong SR*)
```

The Coq implementation

Of `Ds`

- Let us define some `Ds`:

```
a: ∀ {x y z w}, gs x y z SG w → gp cp_x y z SG w
the: ∀ {x y z u w}, gs x y z u w → gp cp_x y z u w
this: ∀ {x d w}, gs cs_s x d SG w → gp cp_x x d SG w
these: ∀ {x d b w}, gs cs_p x d b w → gp cp_p x d PLR w
```

- The `Ds` have function types, with arguments in {. . .} implicit (implicitly applied). `gs _ _ _ _ _` and `gp _ _ _ _ _` are some compound types (in this case, function applications), so e.g. `a` is a function from `gs _ _ _ SG _` to `gp cp_x _ _ SG _`. In Coq, `_` marks any admissible term or type, and `SG` stands for singular, i.e. `a` takes arguments in singular and returns phrases in singular.

The Coq implementation

Of universals

- In Coq we can also define universal notations, e.g.

```
Parameter D: ∀ {x y z u w}, gs x y z u w → gp cp_x y z u w.
(*universal "D" declared as a variable*)
Notation D’ := (_: gs _ _ _ _ _ → gp cp_x _ _ _ _).
(*universal "D’" defined as a notation*)
```

- The universality of `D` and `D'` comes from `x`, `y`, `z`, `u`, `w` and `_` standing for any admissible term or type, whence the applicability of `D` and `D'` whenever one of `a`, `the`, `this`, `these` applies (an ex. with *the man knows a few men*):

```
Check PRES know (the man) (a (few (PL man))).
Check TAM know (D’ man) (D (Q (PL man))).
```

Conclusion

Universal Grammar

- We have shown how to build an extensive and robust substantive UG using supercategories (categories of categories)
- In particular, we proposed the following approach to UG:
  - The universality of categories `PN`, `CON`, `XP`, `S`, `IS` and `CONC`
  - The universality of supercategories `CA`, `CAP`, `Q`, `D`, `TAM`, `POS`, `ADL`, `R` and `X`

Conclusion

Type-theoretical modeling

- The main contributions:
  - An integrated modeling of syntax, morphology and compositional semantics (in the form of selectional restrictions).
  - To account for systematic violations of selectional restrictions by metaphor and metonymy, we have shown how to model them type-theoretically.
  - An implementation of a fragment of NL and UG in the Coq proof assistant (`https://gitlab.com/eluuk/nlc`).
  - The implementation shows how to model many (super)categories of NL, some of them universal, in a purely functional type system (i.e. one comprising only functions and their types) with dependent and polymorphic types.
  - It seems likely that the underlying formalism could be also encoded in a simpler type system, which remains a future work.

References I

And thanks

Asher, N. (2014).
Selectional restrictions, types and categories.
Journal of Applied Logic, 12(1):75–87.

Chomsky, N. (1970).
Remarks on nominalization.
In Jacobs, R. and Rosenbaum, P., editors, *Readings in English Transformational Grammar*, pages 184–221. Ginn, Waltham.

Chomsky, N. (1981).
Lectures on Government and Binding.
Foris, Dordrecht.

Chomsky, N. (1995).
The Minimalist Program.
Current studies in linguistics series 28. MIT Press, Cambridge, MA.

Evans, N. and Levinson, S. C. (2009).
The myth of language universals: Language diversity and its importance for cognitive science.
Behavioral and Brain Sciences, 32:429–492.

Greenberg, J. H. (1966).
Some universals of grammar with particular reference to the order of meaningful elements.
In Greenberg, J. H., editor, *Universals of Grammar*, pages 73–113. MIT Press, Cambridge, MA.

References II

And thanks

Haspelmath, M. (2007).
Pre-established categories don't exist: Consequences for language description and typology.
Linguistic Typology, 11(1):119–132.

Jackendoff, R. (2002).
Foundations of language: brain, meaning, grammar, evolution.
Oxford University Press, Oxford, New York.

Luo, Z. (2010).
Type-theoretical semantics with coercive subtyping.
In *Semantics and Linguistic Theory*, volume 20, pages 38–56, Vancouver.

Luuk, E. (2010).
Nouns, verbs and flexibles: implications for typologies of word classes.
Language Sciences, 32(3):349–365.

Plank, F., Mayer, T., Mayorava, T., and Filimonova, E. (2009).
The Universals Archive.

Ranta, A. (2006).
Type Theory and Universal Grammar.
Philosophia Scientiæ. Travaux d'histoire et de philosophie des sciences, (6):115–131.