Logic, Semantics, Universal Grammar, Translation

Aarne Ranta

May 8, 2010

Logic and Language Seminar, Gothenburg, 7 May 2010

Based on material from

Nordiskt Filosofimöte: Logik, filosofi och språk 45 år senare, Helsingfors, October 18, 2002

Type Theory Workshop, Leyden, February 6, 2004

"Type Theory and Universal Grammar", *Philosophia Scientiae*, Vol. 6 (Special issue on constructivism), 2006.

MOLTO: Multilingual On-Line Translation. Kick-Off Presentation, Barcelona, March 10, 2010.

A Quiz

von Wright's textbook in analytic philosophy has the original Swedish title

> Logik, filosofi och språk.

How do you translate this into English?

Language vs. Languages

Språk is the Swedish word for "language".

As a noun of the fifth declension, its plural is the same as the singular.

Therefore, the title has two possible English translations:

> Logic, Philosophy, and Language,
>
> Logic, Philosophy, and Languages.

The second translation hardly ever occurs to anyone as a possibility!

Universal grammar

The singular translation is justified: von Wright's book is not about Swedish, English, or Finnish, but **universally** about language.

The idea of speaking universally about language was fundamental in the **universal grammar** notion in mediaeval philosophy.

Two famous quotes

(From Gilson (1922), and later requoted in Lyons (1968))

> Grammar is substantially the same in all languages, even though it may undergo in them accidental variations.
> (Roger Bacon, 13th century)
>
> He who knows grammar in one language, also knows it in another as far as the essentials are concerned. The fact that he cannot, however, speak another language, or understand those who speak it, arises from the difference of words and their formations, which is accidental to grammar.
> (Anonymous, 12th century)

Critics: past and present

Renaissance time by scholars: Alexander Hegius and Erasmus.

American structuralism: Sapir Whorf hypothesis.

Contemporary linguists: the pejorative notion of "armchair linguist", who does not see the diversity of languages.

The Sapir Whorf hypothesis

> No two languages are ever sufficiently similar to be considered as representing the same social reality. The worlds in which different societies live are distinct worlds, not merely the same world with different labels attached
>
> (Sapir 1929)

An easy argument

Languages were said to differ only as for "words and their formations".

A quick translation experiment shows a sense in which this cannot be true:

talossanikin

(one word in Finnish) is translated in English as a four-word utterance,

also in my hoúse

Moreover, the stress on the word *house* is important, since

also in mý house

is translate by *two* words in Finnish:

minunkin talossani.

But we'll see later that the idea that languages only differ as for "words and their formations" does make some sense, after all.

Universal language and Descartes

Descartes, in a letter to Mersenne 1629, told about a universal grammar and dictionary that someone had proposed, with the promise that

> anyone who learns this (universal) language, would also know all the others as dialects of it

Descartes found the idea naïve.

Full letter: `http://www.autodidactproject.org/other/descartes-lg1.html`

But he had his own suggestion: a universal language that would

establish an order among all thoughts . . . in the same way as there is a natural order among numbers, and as one can learn in one day the names of all numbers up to infinity and write them in an unknown language, even though they are an infinity of different words

the invention of this language depends on the true philosophy; for it is impossible otherwise to denumerate all thoughts of men and order them, or even distinguish them into clear and simple ones

if anyone had well explained which are the simple ideas . . . of which all that they think is composed . . . then I would dare to hope for a universal language easy to learn, pronounce, and write and . . . which would help judgement, representing all things to it so distinctly that error would be almost impossible

Universal language and Leibniz

More well known than Descartes's suggestion: Leibniz's *characteristica universalis* (1732):

> a symbolic language permitting mechanized reasoning by means of a *calculus ratiocinator*
>
> mathematical notation such that the elements of the notation correspond to the elements of things and facts

Not sure whether Leibniz thought of a bridge between different languages.

Leibniz's and Descartes's idea today

The main aspect: a *calculus* can replace creative reasoning.

Contemporary variant: the universal notation can be manipulated by a *computer program*.

Such a program can

- decide the correctness of judgements
- translate between languages

Two dimensions of universality

Two senses of universality:

horizontal universality: generality across languages

vertical universality: generality across subject matters

These aspects are *orthogonal*.

To assess any argument for or against a "universal grammar", we have to find out which sense is meant.

Roger Bacon: horizontal universality.

Leibniz: vertical universality.

Descartes: both horizontal and vertical.

The Sapir Whorf hypothesis: horizontal criticism

Recall:

> No two languages are ever sufficiently similar to be considered as representing the same social reality. . .

Often discussed as a problem of **translation** between languages:

> Not only is it difficult to *find* a translation from one language to another, but it may even happen that there *is* no translation, since the content expressed by the source language utterance has no counterpart in the target language.

Gödel incompleteness: vertical criticism

The **incompleteness** theorem of Gödel (1931) implies:

> There cannot be any formal system that is complete for all mathematics—let alone for all subject matters.

Wittgestein: vertical criticism

Wittgenstein's late philosophy (1953):

> there is no such thing as language, but just a collection of **language games**
>
> there is no such a thing as the meaning of a word (*simpliciter*), but only its uses in different language games.

An individual language game is a unit that has a set of rules that can possibly be formalized into a formal system; but the totality of language games cannot be formalized.

The main thesis of this talk:

> we can achieve horizontal universality but not vertical universality.

In other words, we can build cross-linguistic grammars on limited domains, or, "we can translate language games".

Cross-linguistic language game

A domain of multilingual activity and a tradition of translation, e.g.

- among scientists within one discipline,
- among employees within an industrial corporation,
- among sportsmen practicing the same sport.

Cross-linguistic equivalence

Equivalence based on language game $\neq$
genetic relatedness of languages

> Swedish (Germanic) and Finnish (Fenno-Ugrian) are largely intertranslatable—the bilinguality legislation in Finland says that all official documents must exist in both languages.
>
> The two Germanies after the Second World War were reported to be drifting apart linguistically.

The translatability criterion

To test horizontal equivalence betwen languages A and B, find out if

> it is possible to **translate** between A and B

This may fail in practice for some technical reason; what we mean is we can replace this criterion by "translatability in principle":

> it is possible to **express the same things** in A and B.

Difficulties in translatability

In a sense, translatability fails very soon:

- the ambiguity of *språk* in von Wright's book title does not translate to English
- in the Bible, there are so-called alphabetical Psalms, where subsequent verses begin with subsequent letters of the Hebrew alphabet; translators have had different ambition levels in reproducing this feature

Abstraction levels for translatability

Normally, we speak of translation on some **level of abstraction**.

A typical level is **semantical**:

> the translation of expressions **as expressions for certain things**

To define the semantic abstraction level:

> to take the things expressed as starting point—not the languages!

I.e. start with a mathematical model of those things and see how it is reflected in languages.

Do *not* start with a language and look for a model for the language.

Logical formalization

Formalization exercises in elementary logic teaching: translate natural language sentences into logical formulae, e.g.

$$\textit{every man loves Mary}$$

is formalized as

$$(\forall x)(\textit{man}(x) \supset \textit{love}(x, \textit{Mary}))$$

This gives a painful awareness that something is getting lost in translation!

It is much less painful to start from the formula and ask how to express it in natural language.

Curry: tectogrammatical vs. phenogrammatical structure

Curry (1963):

> **tectogrammatical structure**: how expressions are divided into meaningful parts
>
> **phenogrammatical structure**: how expressions look like

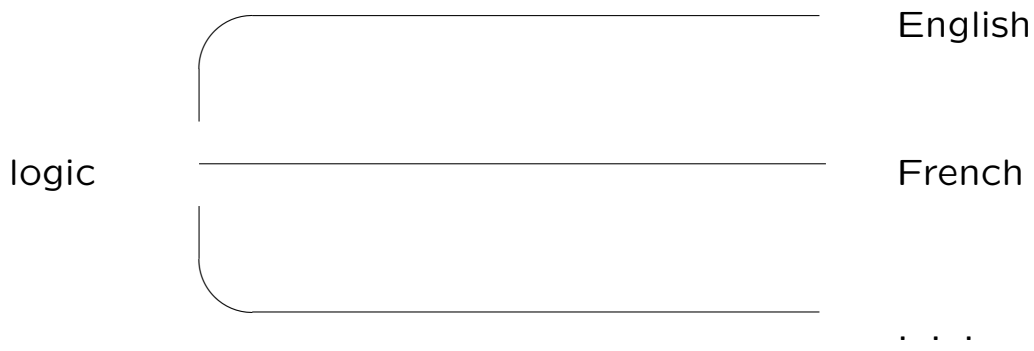
Semantics applies to tectogrammatical structure.

Subsequent verses beginning with subsequent letters belongs to the phenogrammatical structure of a Psalm.

Curry's model of multilingual grammar

tectogrammatical structure

phenogrammatical structure



The implementation of Curry's idea

Montague (1974): the semantic aspect of Curry's model (Dowty 1982).

Shaumyan (1965) and Desclés (1994): the multilingual aspect.

Rosetta project(1995): machine translation based on Montague grammar.

Vertical universality and Curry's idea

In Curry: not clear.

Montague: the semantics of English consists of giving truth conditions to all English sentences expressible in higher-order predicate calculus with modal operators.

Grammatical Framework

GF = Grammatical Framework (Ranta 1998–)

Language-independent representation level: the **Constructive Type Theory** of Martin-Löf.

To this, GF adds forms of judgement to express **linearization rules**.

Abstract vs. concrete syntax

Computer science terminology dating back to Landin (1967, "The next 700 programming languages")

Example: variable declarations in Pascal and C have different concrete syntax but the same abstract syntax

```
var x : Integer
```

```
int x
```

Abstract syntax level: semantics, type checking, evaluation

Concrete syntax level: printing, parsing

Curry's model in GF

abstract syntax

concrete syntax

The framework idea

Logical Framework: type theory used for describing and implementing individual logical calculi and mathematical theories.

Example (a fragment of propositional calculus):

```
cat Prop ;
cat Proof Prop ;

fun Conj : Prop -> Prop -> Prop ;
fun ConjI : (A, B : Prop) ->
                Proof A -> Proof B -> Proof (Conj A B) ;
```

There need not be one unifying theory of mathematics in terms of which everything is defined (such as set theory).

Logical Frameworks in practice

Gain: operations such as inference rule application and variable binding are defined and implemented once and for all, on the **framework level**.

Works for both constructive and classical mathematics: de Bruijn's AUTOMATH is an early example (1967) for classical mathematics.

Systems: LF, Coq, ALF/Alfa, LEGO,...

GF judgement forms

Abstract syntax, inherited from LF:

```
cat C G        -- C is a category in context G
fun f : A      -- f is a function of type A
```

Concrete syntax, for each category C and function f:

```
lincat C = T   -- category C has the linearization type T
lin f = t      -- function f is linearized as t
```

GF grammars

A **grammar** is a pair

$$< \mathcal{A}, \mathcal{C} >$$

of abstract and concrete syntax. A **multilingual grammar** is a pair

$$< \mathcal{A}, \{\mathcal{C}_1, \ldots, \mathcal{C}_n\} >$$

with a set of concrete syntaxes sharing an abstract syntax.

Linearization and parsing

Linearization takes a tree in abstract syntax to an object in the corresponding linearization type (e.g. a string).

Parsing takes a string into a set of abstract syntax trees. (Empty set: parse fails; more than one elements: ambiguity).

Translation from $\mathcal{C}_i$ to $\mathcal{C}_j$ is parsing in $\mathcal{C}_i$ followed by linearization in $\mathcal{C}_j$.

Implementation gain: both linearization and parsing algorithms are defined for all grammars on the framework level.

Example: a fragment of arithmetic, abstract syntax

```
cat Nat ;
cat Prop ;

fun Zero : Nat ;
fun Even : Nat -> Prop ;
```

Arithmetic, English concrete syntax

```
lincat Nat = {s : Str} ;
lincat Prop = {s : Str}

lin Zero = {s = "zero"} ;
lin Even x = {s = x.s ++ "is" ++ "even"} ;
```

Translating between English and French: the goal

We need French expressions in different forms: feminine, masculine, indicative, subjunctive.

zero is even
zéro est pair

the sum of zero and zero is even
la somme de zéro et de zéro est paire

there exists an x such that x is even
il existe un x tel que x soit pair

Parameter type definitions

We need to introduce anothe judgement form in concrete syntax, to define **parameter types**:

```
param Mod = Ind | Subj ;
param Gen = Masc | Fem ;
```

These judgements are similar to inductive datatype definitions.

Arithmetic, French concrete syntax

```
lincat Nat = {s : Str ; g : Gen} ;
lincat Prop = {s : Mod => Str} ;

lin Zero = {s = "zéro" ; g = Masc} ;
lin Even x = {s =
    table {
      m => x.s ++
           case m   of {Ind  => "est" ;  Subj => "soit"} ++
           case x.g of {Masc => "pair" ; Fem  => "paire"}
    }
  } ;
```

Abstraction over morphosyntactic variations

Translations between English and French can now be obtained via abstract syntax.

The abstract syntax makes an **abstraction over morphosyntactic variations**.

Abstraction over word order

Needed in translations betwen English and German:

if zero is even
Null ist gerade

if zero is even then one is odd
wenn Null gerade ist, dann ist Eins ungerade

Arithmetic in German

```
param Ord = Main | Inv | Sub ;
lincat Prop = {s : Ord => Str} ;

lin Even x = {s =
    table {
      Main => x.s ++ "ist" ++ "gerade" ;
      Inv  => "ist" ++ x.s ++ "gerade" ;
      Sub  => x.s ++ "gerade" ++ "ist"
      }
   } ;
```

Precision in translation: variable binding

Binding. In context-free grammar, English

> *there exists an x such that x is even and x is prime*

is ambiguous between two syntax trees, linearized in French

> *il existe un x tel que x soit pair et que x soit premier*
> *il existe un x tel que x soit pair et x est premier*

The latter analysis can be rejected since it has a dangling variable.

Precision in translation: type checking

Ex. 2: pronoun type. The English pronoun *it* has three possible translations in German: *er*, *sie*, *es*. The sentence

if the function f has a maximum, then it reaches it at 0

has two occurrences, which are resolved by the type of the predicate *reach*:

wenn die Funktion f ein Maximum hat, dann reicht sie es bei 0

Precision in translation: context knowledge

English original:

> *When leaving the building: erase your name from the board. If the board is empty, enter the code and then leave the premises. If the board is not empty, just leave the premises.*

German translation obtained by an alarm system grammar in GF:

> *Beim Ausgang vom Gebäude: beseitigen Sie Ihren Namen von der Tafel. Wenn die Tafel leer ist, führen Sie die Codezahl ein und verlassen Sie die Räume. Wenn die Tafel nicht leer ist, verlassen Sie nur die Räume.*

German translation obtained by the generic Systran translator:

> *Wenn das Gebäude gelassen: löschen Sie Ihren Namen vom Brett. Wenn das Brett leer ist, eingeben Sie den Code n und lassen Sie die Voraussetzungen. Wenn das Brett nicht leerer, gerader Urlaub die Voraussetzungen ist.*

Context knowledge $\approx$ knowledge of language game

The unity of a language endangered

If there is no common language game, can there be common structure?

Example:

zero is even

John is old

the weather is beautiful

Is there a common structure of adjectival predication?

If only logical structure exists, can we even have expected common structures in *one* language game?

Example: the indefinite article

John is standing by a stove (existential over stoves)

Arthur Martin is a gas stove (typing judgement)

John is baking a cake (factive)

Cf. de Bruijn, "Het onbepaalde lidwoord in wiskundige Nederlands" (1978).

Resource grammars

Raise the level of abstraction in concrete syntax:

Define linearization as translation into trees in resource grammar.

The resource grammar aims to complete linguistic description of a language, independent of application.

No semantics — it is given in **application grammars**.

Example resource grammar rules for English

```
cat S ;    -- sentence
cat CN ;   -- common noun
cat NP ;   -- noun phrase
cat VP ;   -- verb phrase
cat Adj ; -- adjective

fun PredVP  : NP -> VP -> S ;
fun PredAdj : Adj -> VP ;
fun Indef   : CN -> NP ;
```

```
param Number = Sg | Pl ;
param Person = P1 | P2 | P3 ;

lincat CN = {s : Number => Str} ;
lincat NP = {s : Str ; n : Number ; p : Person} ;
lincat VP = {s : Number => Person => Str} ;

lin PredVP np vp = {s = np.s ++ vp.s ! np.n ! np.p} ;
lin PredAdj adj  = {s = \\n,p => verbBe ! n ! p ++ adj.s} ;
lin Indef cn     = {s = artIndef ++ cn.s ! Sg ; n = Sg ; p = P3} ;
```

Using resource grammar

In English arithmetic:

```
lincat Prop = S ;
lincat Nat = NP ;

lin Even x = PredVP x (PredAdj adjEven) ;
```

Resource lexicon

Words with their morphological properties

```
fun adjEven : Adj ;
lin adjEven = {s = "even"} ;
```

This can be used for different semantic purposes, e.g.

- the divisibility of a number by 2
- the smoothness of a surface

The indefinite article:

```
fun Exist  : Domain -> Quantifier ;
fun Typing : Ident -> Domain -> Judgement ;
fun Bake   : BakeObject -> Action ;

lin Exist A    = Indef A ;
lin Typing a A = PredVP a (Indef A)) ;
lin Bake b     = ComplV2 verbBake (Indef b) ;
```

French resource grammar: concrete syntax

```
lincat
  S  = {s : Mod => Str} ;
  NP = {s : Str ; g : Gender ; n : Number ; p : Person} ;
  VP = {s : Mod => Gender => Number => Person => Str} ;

lin PredVP np vp =
  {s = \\m => np.s ++ vp.s ! m ! np.g ! np.n ! np.p} ;
```

Abstract syntax is shared with English!

German resource grammar: concrete syntax

```
lincat VP = {s,s2 : Number => Person => Str} ;

lin PredVP np vp =
  let
    subj  = np.s ! Nom ;
    verb  = vp.s  ! np.n ! np.p ;
    compl = vp.s2 ! np.n ! np.p
  in
    {s = table {
       Main => subj ++ verb ++ compl ;
       Inv  => verb ++ subj ++ compl ;
       Sub  => subj ++ compl ++ verb
       }
    } ;
```

Using resource grammar in French and German arithmetic

```
lincat Prop = S ;
lincat Nat = NP ;

lin Even x = PredVP x (PredAdj adjPair) ;

lincat Prop = S ;
lincat Nat = NP ;

lin Even x = PredVP x (PredAdj adjGerade) ;
```

When concrete syntax is defined in terms of resource grammar, only the lexicon part of the resource is different in different languages.

Recall:

> *He who knows grammar in one language, also knows it in another as far as the essentials are concerned. The fact that he cannot, however, speak another language, or understand those who speak it, arises from the* **difference of words and their formations**, *which is accidental to grammar.*

There are counterexamples: to translate

x misses y

from English to French, we flip the arguments:

y manque à x

Only the abstract syntax predicate `Miss` can be shared: concrete syntax uses different structures:

```
lin Miss x y = PredVP x (ComplV2 verbMiss y)    -- English
lin Miss x y = PredVP y (ComplV2 verbManquer x) -- French
```

Syntax independent of semantics?

Famous quote from Montague:

> I fail to see any interest in syntax except as a preparation for semantics.

Our objection: this is not true of resource syntax.

Conclusion

We can achieve horizontal universality in semantically delimited domains.

We can even achieve vertical universality, within an individual language on the concrete syntax level.

	English resource	French resource	Italian resource
sailing			
mathematics			
arithmetic			
geometry			