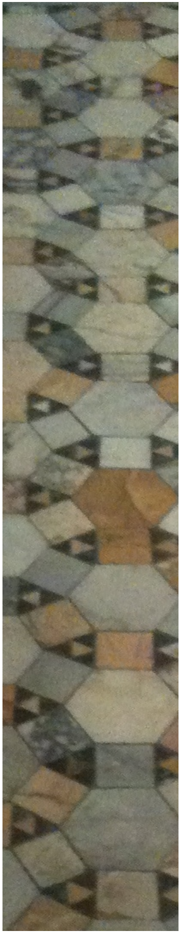


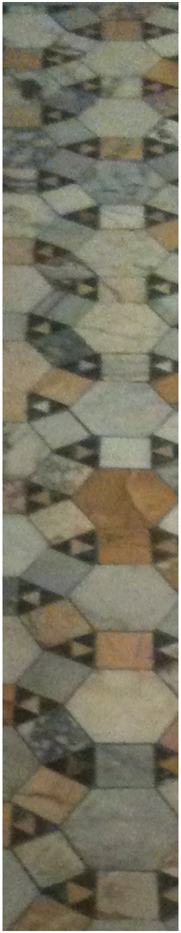
Aristotle’s square of opposition and Hilbert’s epsilon: some linguistic remarks

Christian Retoré

Université de Bordeaux LaBRI (& IRIT Melodi, Toulouse)

4th world congress on the Square of Oppostion

Roma, 5-9 Maggio 2014



A Computing logical forms (à la Montague)

A.1. Logic

Logic, philosophy of language, semantics...

Difficult to tell the difference!

From the beginning two related parts:

- nowadays called **lexical** semantics: interpreting terms (words, noun phrases, even **quantified** nouns phrases)
- nowadays called **formal**/**compositional** semantics: interpreting propositions, reasoning

There is a link between the two.

Quantifcation if concerned by this link.

A.2. Computing logical forms à la Montague

Mind that there are TWO logics: composition / logical form:

One for expressing meanings:
formulae of first or higher order logic, single or multi sorted.

One for meaning assembly:
proofs in intuitionistic propositional logic, λ-terms expressing the well-formedness of formulae.

A.3. Representing formulae within lambda calculus — connectives

Assume that the base types are
$\mathbf{e}$ (individuals, often there is just one) and
$\mathbf{t}$ (propositions)
and that the only constants are
the logical ones (below) and
the relational and functional symbols of the specific logical language (on the next slide).

Logical constants:

- $\sim$ of type $\mathbf{t} \rightarrow \mathbf{t}$ (negation)
- $\supset, \&, +$ of type $\mathbf{t} \rightarrow (\mathbf{t} \rightarrow \mathbf{t})$
  (implication, conjunction, disjunction)
- two constants $\forall$ and $\exists$ of type $(\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t}$

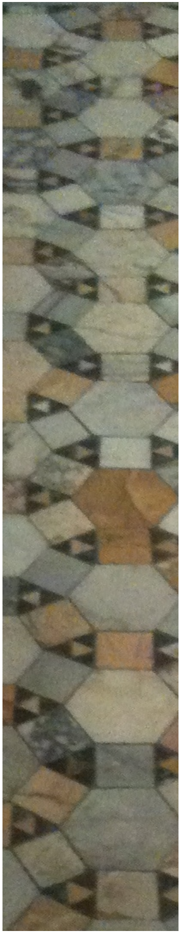
A.4. Representing formulae within lambda calculus — language constants

The language constants for multi sorted First Order Logic:

- R_q of type $\mathbf{e} \rightarrow (\mathbf{e} \rightarrow (.... \rightarrow \mathbf{e} \rightarrow \mathbf{t}))$
- f_q of type $\mathbf{e} \rightarrow (\mathbf{e} \rightarrow (.... \rightarrow \mathbf{e} \rightarrow \mathbf{e}))$

one two-place predicate		
likes	$\lambda x^{\mathbf{e}} \lambda y^{\mathbf{e}}\ (\underline{likes}^{\mathbf{e}\rightarrow(\mathbf{e}\rightarrow\mathbf{t})}\ y)\ x$	
two one place predicates		
cat	$\lambda x.\underline{cat}^{\mathbf{e}\rightarrow\mathbf{t}}$	
sleeps	$\lambda x.\underline{sleep}^{\mathbf{e}\rightarrow\mathbf{t}}$	
two proper names		
Evora	$\underline{Evora} : \mathbf{e}$	*possibly* $(\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t}$
Anne−Sophie	$\underline{Anne-Sophie} : \mathbf{e}$	

Normal terms (preferably η-long) of type $\mathbf{t}$ are formulae.



A.5. Montague semantics. Syntax/semantics.

(Syntactic type)*	$=$	Semantic type	
S^*	$=$	t	a sentence is a proposition
np^*	$=$	e	a noun phrase is an entity
n^*	$=$	$e \to t$	a noun is a subset of the set of entities
$(A\backslash B)^* = (B/A)^*$	$=$	$A \to B$	extends easily to all syntactic categories of a Categorial Grammar e.g. a Lambek CG

A.6. Montague semantics. Algorithm

1. Replace in the lambda-term issued from the syntax the words by the corresponding term of the lexicon.
2. Reduce the resulting λ-term of type t its normal form corresponds to a formula, the ”meaning”.

A.7. Ingredients: a parse structure & a lexicon

Syntactical structure

(some (club)) (defeated Leeds)

Semantical lexicon:

word	***semantics : λ-term of type (sent. cat.)**** x^{v} *the variable or constant x is of type v*
some	$(e \to t) \to ((e \to t) \to t)$ $\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}(P\ x)(Q\ x))))$
club	$e \to t$ $\lambda x^{e}(\mathtt{club}^{e\to t}\ x)$
defeated	$e \to (e \to t)$ $\lambda y^{e}\ \lambda x^{e}\ ((\mathtt{defeated}^{e\to(e\to t)}\ x)y)$
Leeds	e Leeds

A.8. Computing the semantic representation

1) Insert the semantics terms into the parse structure

2) β reduce the resulting term

$$\Big(\big(\lambda P^{e\to t}\, \lambda Q^{e\to t}\, (\exists^{(e\to t)\to t}\, (\lambda x^{e}(\wedge(P\ x)(Q\ x))))\big)\big(\lambda x^{e}(\mathtt{club}^{e\to t}\ x)\big)\Big)$$

$$\Big(\big(\lambda y^{e}\ \lambda x^{e}\ ((\mathtt{defeated}^{e\to(e\to t)}\ x)y)\big)\ Leeds^{e}\Big)$$

$$\downarrow \beta$$

$$\big(\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}(\mathtt{club}^{e\to t}\ x)(Q\ x))))\big)$$

$$\big(\lambda x^{e}\ ((\mathtt{defeated}^{e\to(e\to t)}\ x)Leeds^{e})\big)$$

$$\downarrow \beta$$

$$\big(\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge(\mathtt{club}^{e\to t}\ x)((\mathtt{defeated}^{e\to(e\to t)}\ x)Leeds^{e})))\big)$$

Usually human beings prefer to write it like this:

$$\exists x : e\ (\mathtt{club}(x)\ \wedge\ \mathtt{defeated}(x, Leeds))$$

A.9. Montague: good architecture / limits

Good trick (Church):

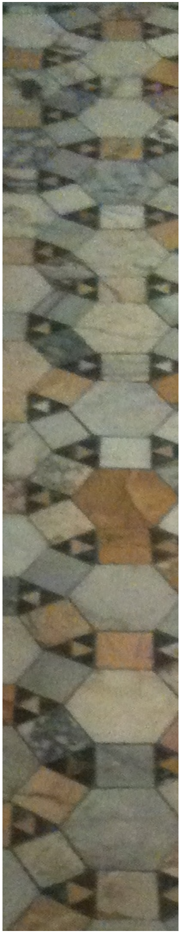
> a propositional logic for meaning assembly (proofs/λ-terms)
>
> computes
>
> formulae of another logic HOL / FOL (formulae/meaning; no proofs)
>
> reification for remaining in FOL can be discussed

The dictionary says ”barks” requires a subject of type ”animal”.
How could we block:

(1) * The chair barked.

By type mismatch, $(f^{A\to X}(u^B))$ hence **many types** are needed.

Description with few operators
$\longrightarrow$ **factorise** similar operations acting on terms/types
$\longrightarrow$ **quantification over types**



B ΛTy_n: a many sorted framework

B.1. System F

Types:

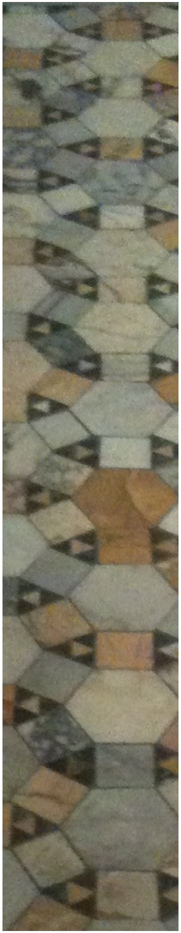
- $\mathbf{t}$ (prop)
- many entity types $\mathbf{e}_i$
- type variables $\alpha, \beta, \ldots$
- $\Pi\alpha.\ T$
- $T_1 \to T_2$

Terms

- Constants and variables for each type
- $(f^{T \to U} a^T) : U$
- $(\lambda x^T.\ u^U) : T \to U$
- $t^{(\Lambda\alpha.\ T)}\{U\} : T[U/\alpha]$
- $\Lambda\alpha.u^T : \Pi\alpha.T$ — no free α in a free variable of u.

The reduction is defined as follows:

- $(\Lambda\alpha.\tau)\{U\}$ reduces to $\tau[U/\alpha]$ (remember that α and U are types).
- $(\lambda x.\tau)u$ reduces to $\tau[u/x]$ (usual reduction).



B.2. Basic facts on system F

Logician / philosopher often ask whether system F is safe?

- We do not really need system F but any type system with quantification over types. F is syntactically the simplest. (polynomial Soft Linear Logic of Lafont is enough)
- Confluence and strong normalisation — requires the comprehension axiom for all formulae of HA_2. (Girard 1971)
- A concrete categorical interpretation with coherence spaces that shows that there are distinct functions from A to B.
- Terms of type $\mathbf{t}$ with constants of mutisorted FOL (resp. HOL) correspond to multisorted formulae of FOL (resp. HOL)
- Possiblilty to have **coercive sub typing** for ontological inclusion (*cats are animals* etc.)

B.3. Examples of second order usefulness

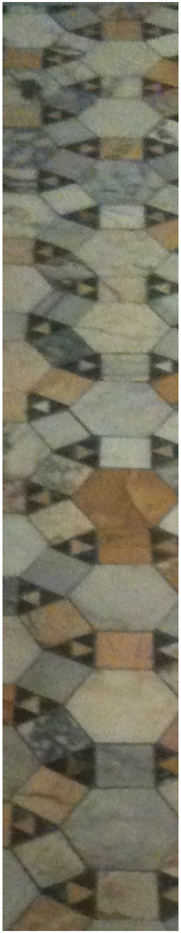
Arbitrary modifiers: $\Lambda\alpha\lambda x^{A} y^{\alpha} f^{\alpha\to R}.((\mathrm{read}^{A\to R\to t}\ x)\ (f\ y))$

Polymorphic conjunction:

> Given predicates $P^{\alpha\to\mathbf{t}}$, $Q^{\beta\to\mathbf{t}}$ over respective types α, β,
> given any type ξ with two morphisms from ξ to α and to β
> we can coordinate the properties P, Q
> of (the two images of) an entity of type ξ:

The polymorphic conjunction $\&^{\Pi}$ is defined as the term

$$\&^{\Pi} = \Lambda\alpha\Lambda\beta\lambda P^{\alpha\to\mathbf{t}}\lambda Q^{\beta\to\mathbf{t}}\ \Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}.\ (\mathrm{and}^{\mathbf{t}\to\mathbf{t}\to\mathbf{t}}\ (P\ (f\ x))(Q\ (g\ x)))$$



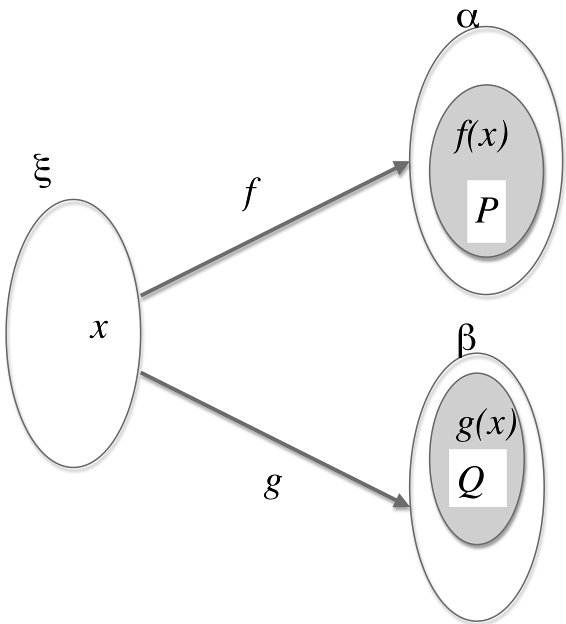


Figure 1: Polymorphic conjunction: $P(f(x))\&Q(g(x))$ with $x:\xi$, $f:\xi\to\alpha$, $g:\xi\to\beta$.

B.4. Types and terms: system F

System F with many base types $\mathbf{e}_i$ (many sorts of entities)

- $\mathbf{e}$ the sort of all entities
- $\mathbf{v}$ events who play a particular role
- *animate*
- *human beings*
- *concepts* ...

$\mathbf{t}$ truth values

types variables roman upper case, greek lower case

usual terms that we saw, with constants (free variables that cannot be abstracted)

Every normal terms of type $\mathbf{t}$ with free variables being logical variables (of a the corresponding multi sorted logic L) correspond to a formula of L.

B.5. The Terms: principal or optional

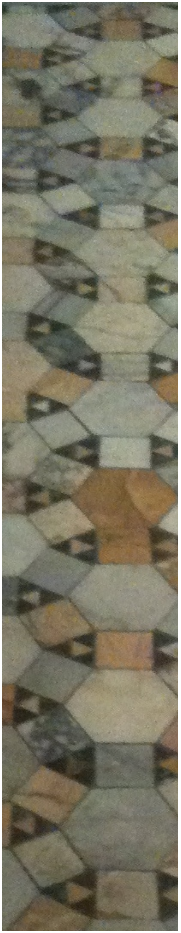
A standard λ-term attached to the main sense:

- Used for compositional purposes
- Comprising detailed typing information (restrictions of selection)

Some optional λ-terms (none is possible)

- Used, or not, for adaptation purposes
- Each associated with a constraint : *rigid,* $\varnothing$

Both function and argument may contribute to meaning transfers.



B.6. RIGID vs FLEXIBLE use of optional terms

RIGID

Such a transformation is exclusive:

the other aspects of the same word are not used.

Each time we refer to the word it is with the same aspect.

FLEXIBLE

There is no constraint.

Any subset of the flexible transformation can be used:

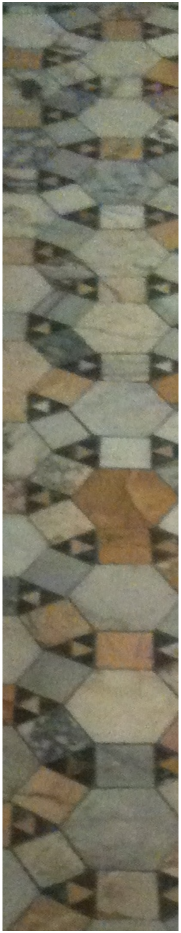
different aspects of the words can be simultaneously used.

B.7. Correct copredication

word	principal λ-term	optional λ-terms	rigid/flexible
Liverpool	$liverpool^T$	$Id_T : T \rightarrow T$	(F)
		$t_1 : T \rightarrow F$	(R)
		$t_2 : T \rightarrow P$	(F)
		$t_3 : T \rightarrow Pl$	(F)
is_spread_out	$spread\_out : Pl \rightarrow \mathbf{t}$		
voted	$voted : P \rightarrow \mathbf{t}$		
won	$won : F \rightarrow \mathbf{t}$		

where the base types are defined as follows:

T	Town
F	football club
P	people
Pl	place



B.8. Meaning transfers

(2) Liverpool is spread out.

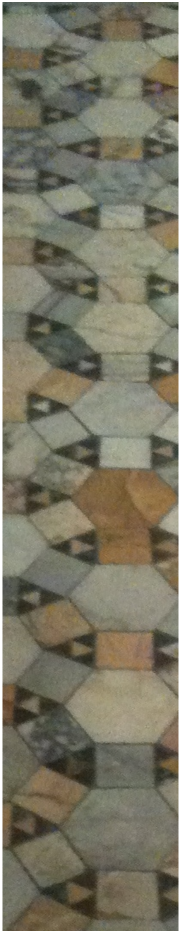
(3) Liverpool won.

(4) Liverpool voted.

$$spread\_out^{Place \rightarrow \mathbf{t}} Liverpool^{Town}$$

Type mismatch, use the appropriate optional term.

$$spread\_out^{Place \rightarrow \mathbf{t}} (t_3^{Town \rightarrow Place} Liverpool^{Town})$$

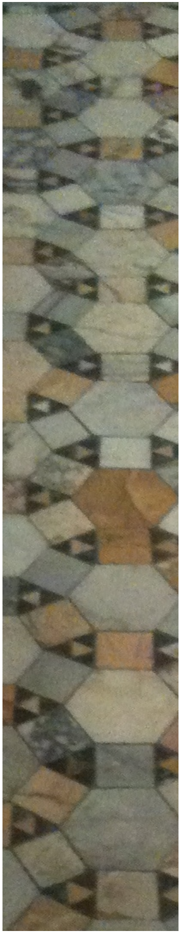


B.9. (In)felicitous copredications

Use polymorphic ”and”... specialised to the appropriate types:

(5) Liverpool is spread out and voted.
Town → *Place* and *Town* → *People*
fine

(6) * Liverpool won and voted.
Town → *FootballClub* and *Town* → *People*
blocked because the first transformation is **rigid**.
(sole interpretation: *football* team or committee voted)



B.10. Liverpool is spread out

Type mismatch:

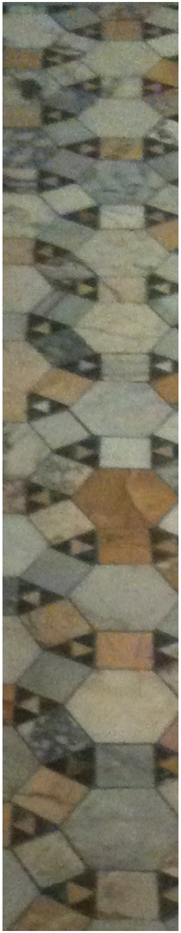
$$spread\_out^{Pl \to \mathbf{t}}(Liverpool^{T})$$

$spread\_out$ applies to “*places*” (type Pl) and not to “*towns*” (T)

Lexicon $t_3^{T \to Pl}$ turns a town (T) into a place (Pl)

$spread\_out^{Pl \to \mathbf{t}}(t_3^{T \to Pl} Liverpool^{T}))$

only one optional term, the (F)/ (R)difference is useless.



B.11. Liverpool is spread out and voted

Polymorphic AND yields: $(\&^{\Pi}(\mathit{spread\_out})^{Pl\to\mathbf{t}}(\mathit{voted})^{P\to\mathbf{t}})$

Forces $\alpha := Pl$ and $\beta := P$, the properly typed term is

$$\&^{\Pi}\{Pl\}\{P\}(\mathit{is\_wide})^{Pl\to\mathbf{t}}(\mathit{voted})^{P\to\mathbf{t}}$$

It reduces to:

$$\Lambda\xi\lambda x^{\xi}\ \lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}(\mathrm{and}^{\mathbf{t}\to\mathbf{t})\to\mathbf{t}}\ (\mathit{is\_wide}\ (f\ \ x))(\mathit{voted}\ (g\ \ x)))$$

Syntax applies it to "*Liverpool*" so $\xi := T$ yielding

$\lambda f^{T\to Pl}\lambda g^{T\to P}(\mathrm{and}\ (\mathit{is\_wide}\ (f\ \ \mathit{Liverpool}^{T}))(\mathit{voted}\ (g\ \ \mathit{Liverpool}^{T}))))$.

The two flexible optional λ-terms $t_2 : T \to P$ and $t_3 : T \to Pl$ yield

$(\mathrm{and}\ (\mathit{is\_wide} Pl \to \mathbf{t}\ (t_3^{T\to Pl}\ \mathit{Liverpool}^{T}))(\mathit{voted}^{Pl\to\mathbf{t}}\ (t_2^{T\to P}\ \mathit{Liverpool}^{T})))$

B.12. Liverpool voted and won

As previously but with *won* instead of *spread_out*.

The term is:
$\lambda f^{T \to Pl} \lambda g^{T \to P} (\mathrm{and}\ (won\ (f\ Liverpool^{T}))(voted\ (g\ Liverpool^{T})))$

for "*won*", we need to use the transformation $t_1 : T \to F$

but T_1 is rigid, hence we cannot access to the other needed transformation into a "*place*".

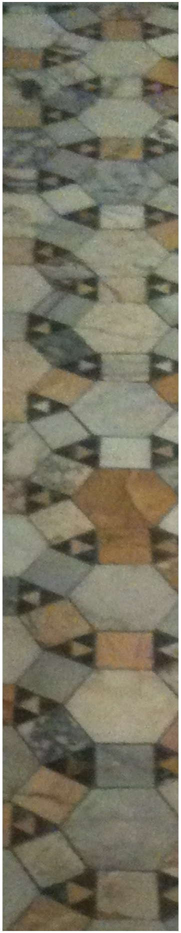
B.13. Other phenomena handled by the same model

Virtual traveller / fictive motion (with Moot & Prévot)
“*The road does down for twenty minutes*”

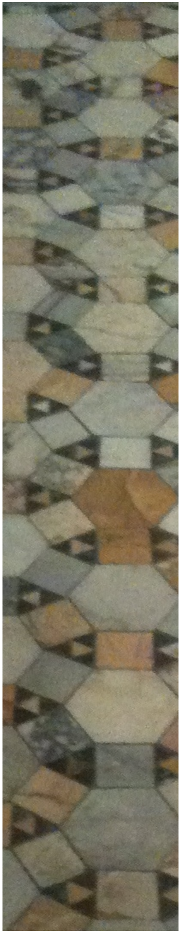
Deverbals: meanings copredications (with Livy Real):
“*A assinatura atrasou três dias / * e estava ilegìvel.*”

Plurals: collective / distributive readings (with Moot)
(The players from) Benfica won although they had the flu.

Generalised quantifiers (“*most*”) with generic elements.
The Brits love France.



C Determiners, quantifiers in the Montagovian generative lexicon



C.1. Usual Montagovian treatment

(1) A tramp died on the pavement.

(2) Something happened to me yesterday.

Usual view (e.g Montague)

Quantifier applies to the predicate,

$[\mathit{something}] = \exists : (\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$

and when there is a restriction to a class: $[\mathit{some}]$

$\lambda P^{\mathbf{e}\to\mathbf{t}} \lambda Q^{\mathbf{e}\to\mathbf{t}} (\exists \lambda x^{\mathbf{t}}.\&(P\ x)(Q\ x)) : (\mathbf{e} \to \mathbf{t}) \to (\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$

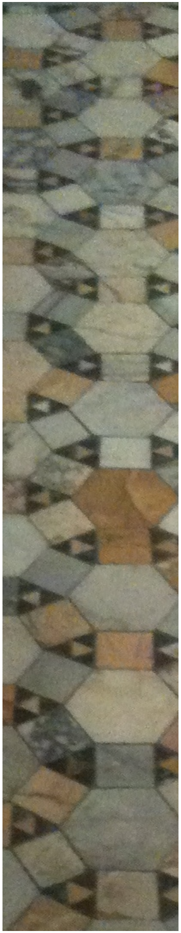
C.2. Quantifier: critics of the standard solution 1/3

Syntactical structure of the sentence $\neq$ logical form.

(7) Orlando di Lasso composed some motets.

(8) syntax (Orlando di Lasso (composed (some (motets))))

(9) semantics: (some (motets)) $\underline{(\lambda x.\ \text{OdL composed } x)}$



C.3. Quantifier: critics of the standard solution 2/3

Asymmetry class / predicate

(10) Some politicians are crooks

(11) ? Some crooks are politicians

(12) $\exists x.\ crook(x) \& politician(x)$

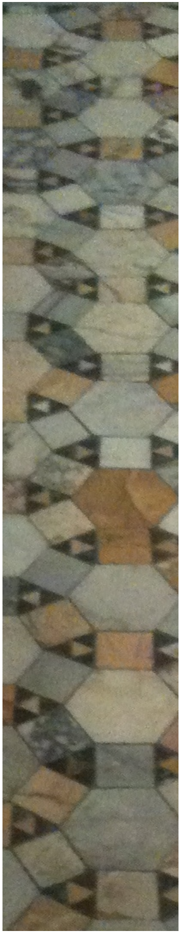
C.4. Quantifier: critics of the standard solution 3/3

There can be a reference before the utterance of the main predicate (if any):

(13) Premier voyage, New-York. (B. Cendrars)

(14) Un luth, une mandore et une viole que Michel-Ange ne sait pas appeler oud, saz, et kaman, accompagnés d'un tambour de basque animé par les doigts tantôt caressants, tantôt violents d'une jeune femme habillée en homme, dont les bracelets de métal tintent en rythme, ajoutent de temps en temps une percussion métallique au concert et distraient un peu l'artiste florentin de cette musique à la fois sauvage et mélancolique. (M. Énard)

(15) Lundi, mercredi et vendredi, une machine de couleurs, mardi et jeudi, une machine de blanc, le samedi, les draps, le dimanche, les serviettes. (Blog)

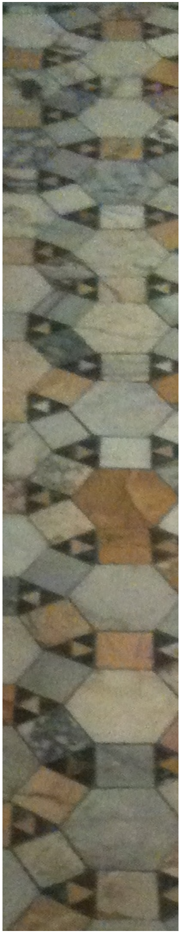


C.5. A solution: Hilbert's epsilon

$$F(\varepsilon_x F) \equiv \exists x.\ F(x)$$

A term (of type individual) $\varepsilon_x F$ associated with F: as soon as an entity enjoys F the term $\varepsilon_x F$ enjoys F.

The operator ε binds the free occurrences of x in F.



C.6. Rules for ε — main properties

Hilbert's work: fine! (Grundlagen der Mathematik, with P. Bernays)

Rule 1: From $P(x)$ with x generic infer $P(\varepsilon_x.\neg P(x)) \equiv P(\tau_x.P(x)) \equiv \forall x\ P(x)$

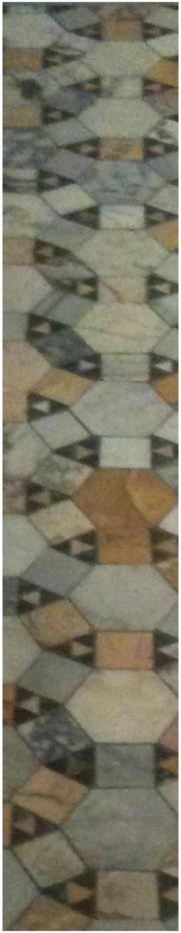
Rule 2: From $P(t)$ infer $P(\varepsilon_x P(x)) \equiv \exists x\ P(x)$

ε-elimination (1st & 2nd ε-theorems), proof of Herbrand theorem.

Little else is known (non standard formulae, full cut-elimination, models), erroneous results cf. Zentralblatt.

$$Sleeps(\varepsilon_x Cat(x)) \equiv ???$$

$$(Cat(\varepsilon_x Cat(x)) \& Sleeps(\varepsilon_x Cat(x))) \equiv \exists x\ Cat(x) \& Sleeps(x)$$



C.7. Admittedly slightly unpleasant

Heavy notation:

$\forall x \exists y P(x,y)$ is $\quad P(\tau_x P(x, \varepsilon_y P(\tau_x P(x,y),y)), \varepsilon_y P(\tau_x P(x,y),y))$

C.8. “Loose” use of ε

Some A are B.

$B(\varepsilon x.\ A(x))$

Not equivalent to an ordinary formula, in particular not equivalent to the standard: $\exists x.\ A \& B(x)$ but

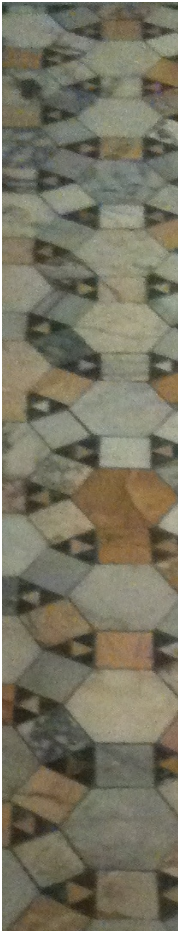
$B(\varepsilon x.\ A(x)) \wedge A(\varepsilon x.\ A(x)) \equiv \exists x.\ B \& A(x)$

Indeed:

$B(\varepsilon x.\ A(x)) \wedge A(\varepsilon x.\ A(x))$
$\vdash B(\varepsilon x.\ B \& A(x)) \wedge A(\varepsilon x.\ B \& A(x))$
$\vdash B \& A(\varepsilon x.\ (B \& A(x))$

Conversely:

$B \& A(\varepsilon x.\ (B \& A(x)) \vdash B(\varepsilon x.\ (B \& A(x)) \vdash B(\varepsilon x.\ B(x))$
$B \& A(\varepsilon x.\ (B \& A(x)) \vdash A(\varepsilon x.\ (B \& A(x)) \vdash A(\varepsilon x.\ A(x))$



C.9. Interpretation

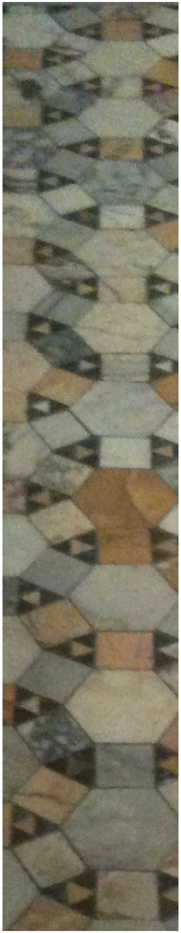
Kind of Henkin witnesses but actually no good interpretation that would entail completeness.

Here is a pleasant intituitive interpretation:

von Heusinger interpretations differ for different occurrences of $\varepsilon_x F(x)$.

(16) a. A tall man went in. A blonde man went out.

b. *Not the same F but necessarily different interpretations.*



C.10. Typed Hilbert operators

Single sorted logic, Frege / Montague style: $\varepsilon : (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{e}$

Many sorted:

$\varepsilon^* : \Lambda\alpha.\ \alpha$

or

$\varepsilon : \Lambda\alpha.\ (\alpha \rightarrow \mathbf{t}) \rightarrow \alpha$

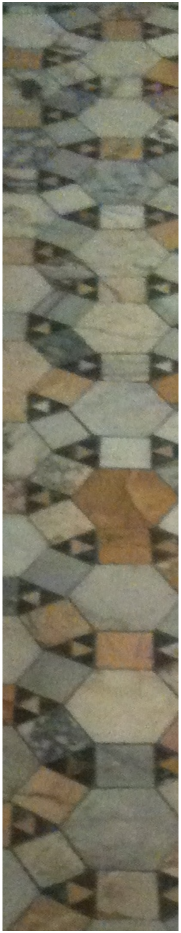
???

either type/formula entails the other:

$$\varepsilon^* = \varepsilon\{\Lambda\alpha.\alpha\}(\lambda x^{\Pi\alpha.\alpha}.\ x\{\mathbf{t}\}) : \Lambda\alpha.\ \alpha$$
$$\varepsilon = \varepsilon^*\{\Lambda\alpha.\ (\alpha \rightarrow \mathbf{t}) \rightarrow \alpha\}$$

ε is more general because type can be mirrored as predicates, but not the converse.

There is no problem of consistency with such constants whose type in unprovable (like fix point Y).



C.11. Intuitive interpretation and logic: some perspectives

Cohabitation of types and formulae of first/higher order logic:

Typing ($\sim$ presupposition) is irrefutable $\mathit{sleeps}(x : \mathit{cat})$

Type to Formula:

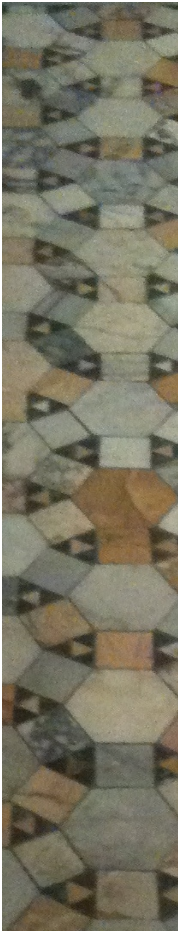
type cat mirrored as a predicate $\widehat{\mathit{cat}} : \mathbf{e} \to \mathbf{t}$

Formula to Type?

Formula with a single free variable $\sim$ type?

$\mathit{cat}(x) \wedge \mathit{belong}(x, \mathit{john}) \wedge \mathit{sleeps}(x) \sim$ type?

At least it is not a natural class.

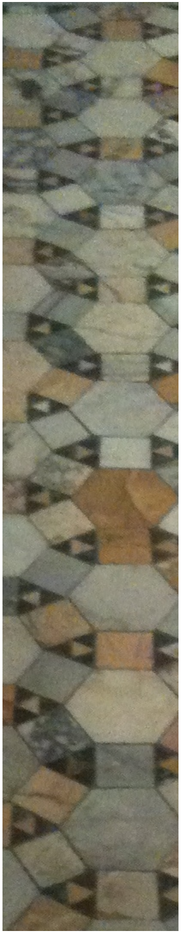


C.12. Computing the proper semantics reading

A cat. cat$^{animal \to \mathbf{t}}$ $(\varepsilon\{animal\}cat^{animal \to \mathbf{t}}) : animal$

Presupposition $F(\varepsilon_x F)$ is added: $cat(\varepsilon\{animal\}cat^{animal \to \mathbf{t}})$

For applying ε to a type say cat,
any type has a predicative counterpart cat (type) $\widehat{cat} : \mathbf{e} \to \mathbf{t}$.
(domains can be restrained / extended)



C.13. Avoiding the infelicities of standard Montague semantics

$\varepsilon_x F$: individual.

1. Can be interpreted as an individual.
2. Follows syntactical structure.
3. Asymmetry subject/predicate.

C.14. E-type pronouns

ε solves the so-called E-type pronouns interpretation (Gareth Evans):

(17) A man came in. He sat dow.

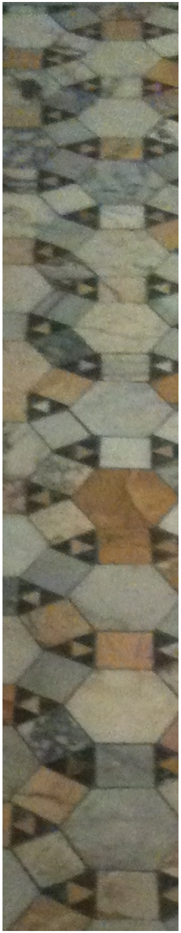
(18) $"He" = "A\ man" = (\varepsilon_x\ M(x))$.

The semantic of a (occurrence of the) pronoun copies the semantic term of its antecedent.

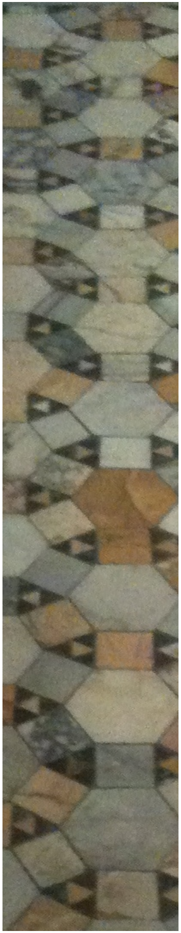
C.15. Difference with choice functions

Choice functions, Skolem symbols:

- One per formula: given one formula one enrich the formal language with a new function symbol and usually, there are no function symbols, when interpreting natural language: as a dictionary, the logical lexicon should be finite.
- No specific deduction system.
- The symmetry problem is still there: it does not go beyond classical logic and the E sentences are still improperly symmetric.



D Other sentences from the square revisited

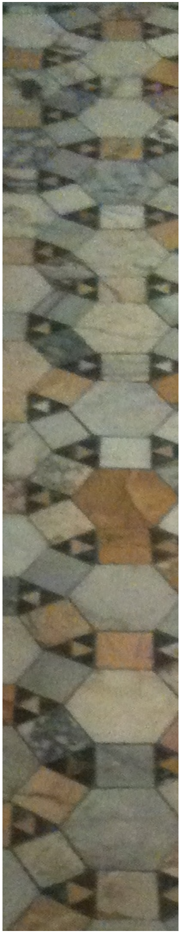


D.1. Formulae of the square

Of course because of the epsilon theorem one can use the translations of standard formula into the epsilon calculus:

- $\forall x.A(x) \Rightarrow B(x) = A(\tau x.\ (A(x) \Rightarrow B(x))) \Rightarrow B(\tau x.\ (A(x) \Rightarrow B(x)))$
- $\exists x.A(x) \& B(x) = A(\varepsilon x.\ (A(x) \& B(x))) \& B(\varepsilon x.\ (A(x) \& B(x)))$
- $\neg \exists x.A(x) \& B(x) = \neg(A(\varepsilon x.\ (A(x) \& B(x))) \& B(\varepsilon x.\ (A(x) \& B(x))))$
- $\neg(\forall x.A(x) \Rightarrow B(x)) = A(\tau x.\ (A(x) \Rightarrow B(x))) \Rightarrow B(\tau x.\ (A(x) \Rightarrow B(x)))$

And there are many more way to write these (this is classical logic).



D.2. E sentences

Clearly E sentences are the most common sentences in natural language.

Existential quantifications structures the discourse as Discourse Representation Theory shows.

For those we proposed as earlier said:

$B(\varepsilon x.\ A(x))$ with typed operators (higher order many sorted logic in type theory).

As said earlier:

D.3. E sentences and definite descriptions

As observed by vo Heusinger, it should be observed that there is little difference between the logical form of definite descriptions and indefinite noun phrase...

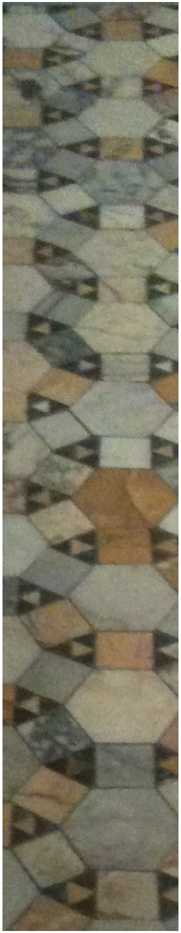
The uniqueness is not always observed.

(19) Recueilli très jeune par les moines de l'abbaye de Reichenau, sur **l'ile du lac de Constance**, en Allemagne, qui le prennent en charge totalement; Hermann étudie et devient l'un des savants les plus érudits du XIème siècle.

The entity may be absolutely new in the discourse.

(20) At this very moment, the donkey entered the lecture all.

It is rather a question of interpretation similar to anaphora.



D.4. A/universal sentences

The correspondence with the E sentence is clear.

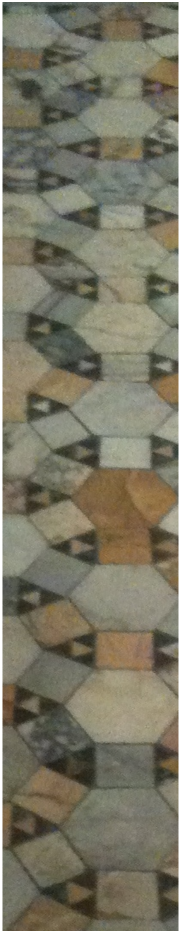
Observed that our setting allow two way to do so (as for the epsilon):

if the noun is a type, the operator should a poly to a type and yields an object of this type: $\Pi\alpha.\ \alpha$

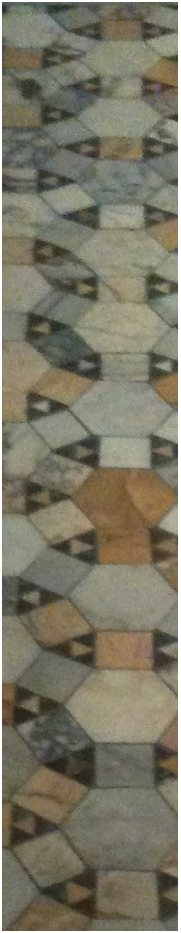
when it is a property the type is $\Pi\alpha.\ (\alpha \to \mathbf{t}) \to \alpha$

D.5. Other sentences: duality

As our understanding of formulae of the epsilon calculus is limited, we prefer to define the other sentences by **negation**: given that epsilon calculus is naturally classical it is endowed with a well behaved negation enjoying De Morgan duality.



E Conclusion



We propose in our type theoretical version of computational semantics some formulation of E sentences (the most common ones) that avoids the usual drawbacks.

It stresses the differences between predicate of one free variable and types.

It open new perspectives on:

- the other formulae A I O of the square of opposition
- the opposition themselves
- underspecified scope: $P(\tau x.\ A(x), \varepsilon x.\ B(x))$
- generalised quantifiers (first work in 2012):

  *If **all** roads lead to Rome, **most** segments of the transportation system lead to Roma Termini!* (Blog: Ron in Rome)