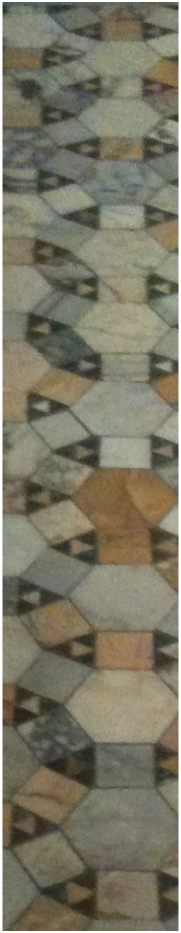


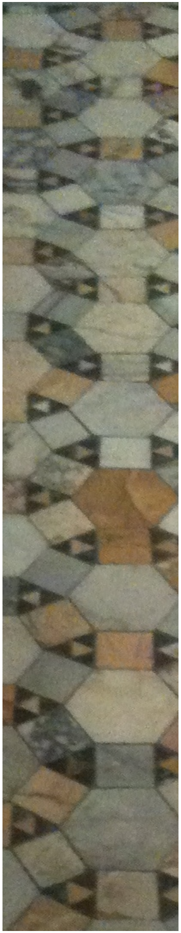
$$\varepsilon$$

L'opérateur epsilon de Hilbert ou comment les éléments génériques permettent de quantifier sans quantificateur.

Christian Retoré

Université de Montpellier, LIRMM

Séminaire GraphiK 29 mai 2015



A Generics and universals

A.1. Ancient and medieval philosophical ideas

A long debated question in logic and metaphysics (from Plato, Aristotle, **Porphyre**, scholastics...)

Universal ”dog” vs. the set of individuals ”dogs”

What is a concept of a dog?

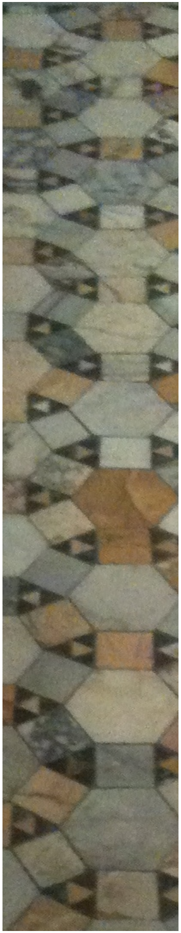
a substance, that exists independently of the individuals falling under this concept

a name without reality, i.e. an abbreviation a word for the class of all individuals falling under the concept

a concept that is a mental construction related to the empirical relation to the set of individuals

A good question (Abélard/Roscelin debate) :

If an illness causes the extinction of all tall dogs, would your concept of dog be altered.



A.2. Ancient and new mathematics

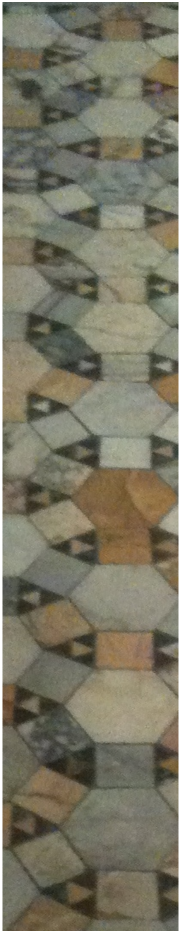
Now let speak about proofs and reasoning on a collection of individuals:

$$\forall x \in \mathscr{C}\, F(x) \equiv \&_{x\in\mathscr{C}} F(x) \equiv P(c_1)\&P(c_2)\&P(c_3)\&P(c_4)\&\cdots$$

How can we prove and refute such a formula? TWO ways:

- We can prove $P(c_1)$ then $P(c_2)$ then $P(c_3)$ then $P(c_4)$... Once we did so for all elements in $\mathscr{C}$ we can perform the conjunction of all these formulae.
- Let x be any element of $\mathscr{C}$...(reasoning) ... $P(x)$ holds. As x does not possess anything special apart from being in $\mathscr{C}$, the property holds for any element in $\mathscr{C}$.

As far as the collection under consideration is finite no difference.



A.3. The dual nature of universal quantification

(1) a. Tout chien a quatre pattes.
b. Le chien a quatre pattes.
c. Les chiens ont quatre pattes.
d. Tous les chiens ont quatre pattes.
e. Chaque chien a quatre pattes.

(2) a. Each dog has four legs.
b. Dogs have four legs.
c. A dog has four legs.

A.4. Distributive readings and individuals

Collection of individuals: cannot accept exceptions, coincidence of properties that can be conjuncted.

- Domain may be complicated:

  "Every one sitting at the table with the uncle of the bride had white shirts."

- Proof by reasoning.

- Refutation: a sentence involving a distributive quantifier on domain *D* can only be refuted by an individual and not by a class unless it is clear that this class intersects the domain *D*.

(3) a. Each bird with both black and white feathers flies.
 b. Not this wound bird.(perfect)
 c. Not autruches. (not good refutation, since the intersection is not obvious)

A.5. Generics NPs and sentences

Generic element: ideal, properties derived by reasoning, can accept exceptions.

- Domain cannot be complicated.

  A person sitting at the table with the uncle of the bride had a white shirt. (cannot mean all of them).

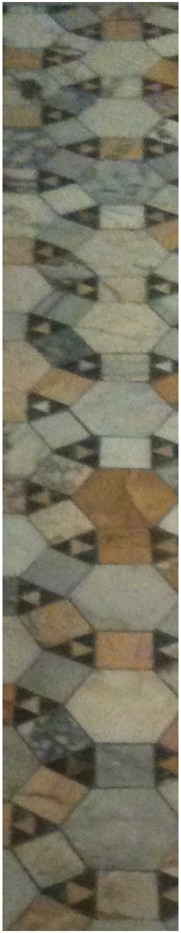
- Refutation:

  The refutation of a sentence involving a generic can only be refuted by another rule: Proof: a proof for each individual.

- (4) a. Birds fly.

  b. Not this wound bird. (not a refutation, generic readings admit exceptions)

  c. Not autruches. (perfect)



A.6. Proof rules with generics — introduction

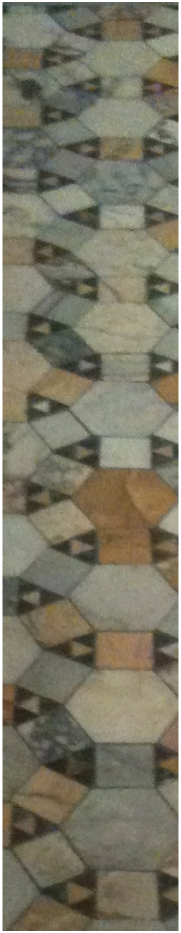
Usual rule when a property has been established for an x which does not enjoy any particular property (i.e. is not free in any hypothesis), one can conclude that the property holds for all individuals:

$$\frac{\begin{array}{c}\text{no free occurrence of } x\\ \text{in any } H_i\\ H_1, \ldots, H_n \vdash P(x)\end{array}}{H_1, \ldots, H_n \vdash \forall x.\ P(x)}\ \forall_i$$

Can be formulated with a generic element:

$$\frac{\begin{array}{c}\text{no free occurrence of } x\\ \text{in any } H_i\\ H_1, \ldots, H_n \vdash P(x)\end{array}}{H_1, \ldots, H_n \vdash P(\tau_x P(x))}\ \forall_i$$

$\tau_x P(x)$ enjoys the property $P(\_)$ when any individual does.



A.7. Proof rules with generics — elimination

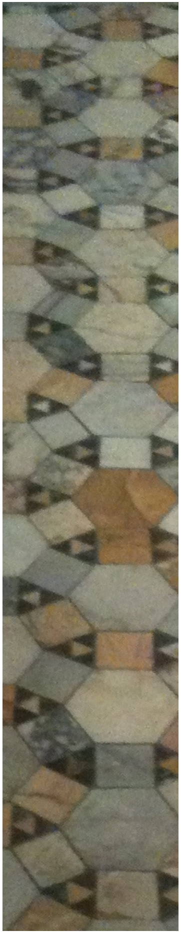
The $\forall$ elimination rule says that when a property has been established for all individuals it can be inferred for any particular terms or individual:

$$\frac{H_1, \ldots, H_n \vdash \forall x.\ P(x)}{H_1, \ldots, H_n \vdash P(a)}\ \forall_e$$

This can be formulated with a generic individual using τ a **subnector** i.e. an operator that builds a term (of type individual) from a formula (Curry's terminology).

$$\frac{H_1, \ldots, H_n \vdash P(\tau_x P(x))}{H_1, \ldots, H_n \vdash P(a)}\ \forall_e$$

If $\tau_x P(x)$ enjoys the property $P(\_)$ then any individual does.

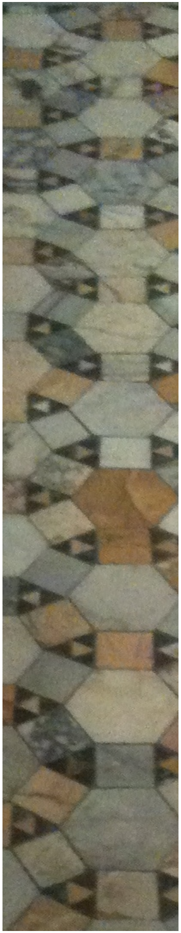


B Hilbert's epsilon

B.1. Russell's iota

As opposed to the generic dog $\tau_x dog(x)$ there is "This dog", "The dog that is sleeping on the sofa,..." the unique individual satisfying P: a term $\iota_x P(x)$.
Russell introduced ι for definite descriptions. It is the ancestor of Hilbert's ε.
A technical problem with ι is that the negation of there exists a unique individual such that P is that there are no such individual or at least two.
As observed by von 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,

(5) 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.

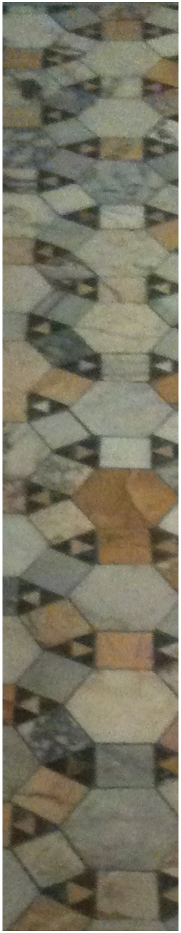


B.2. Hilbert's epsilon

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

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.

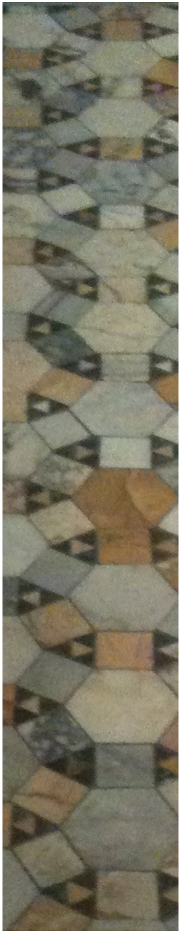
The operator ε binds the free occurrences of x in F.



B.3. Syntax of epsilon in first order logic

Terms and formulae are defined by mutual recursion:

- Any constant in $\mathscr{L}$ is a term.
- Any variable in $\mathscr{L}$ is a term.
- $f(t_1, \dots, t_p)$ is a term provided each t_i is a term and f is a function symbol of $\mathscr{L}$ of arity p
- $\varepsilon_x A$ is a term if A is a formula and x a variable — any free occurrence of x in A is bound by ε_x
- $\tau_x A$ is a term if A is a formula and x a variable — any free occurrence of x in A is bound by τ_x
- $s = t$ is a formula whenever s and t are terms.
- $R(t_1, \dots, t_n)$ is a formula provided each t_i is a term and R is a relation symbol of $\mathscr{L}$ of arity n
- $A \& B$, $A \vee B$, $A \Rightarrow B$, $\neg A$ when A and B are formulae.



B.4. Rules for ε

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

Introduction of the universal quantifier

Rule 1: From $P(x)$ with x generic infer:
$P(\tau_x.P(x))$

Introduction of the existential quantifier:

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

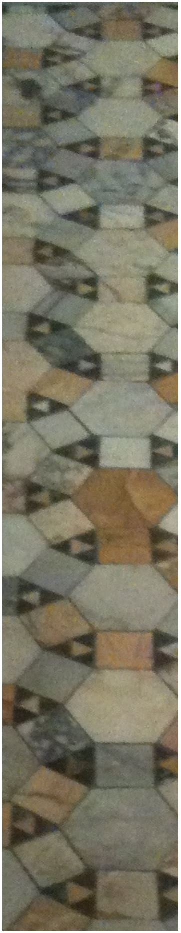
A classical (as opposed to intuitionistic) observation:

$$P(\varepsilon_x P(x)) \equiv \exists x P(x) \equiv \neg\forall x \neg P \equiv \neg\neg P(\tau_x \neg P(x))$$

$$P(\tau_x P(x)) \equiv \forall x P(x) \equiv \neg\exists x \neg P \equiv \neg\neg P(\varepsilon_x \neg P(x))$$

Hence: $\tau_x P(x) = \varepsilon\neg P(x)$ and $\varepsilon_x P(x) = \tau\neg P(x)$

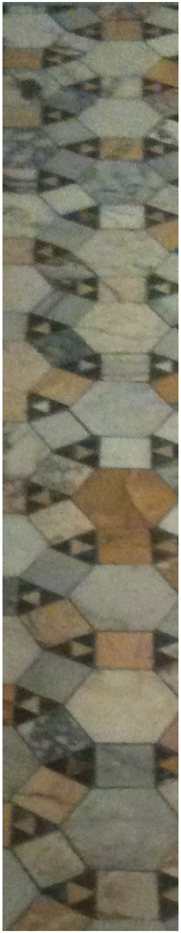
One is enough, usually people chose ε (e.g. Bourbaki in their set theory book).



B.5. Relation to first order logic

The quantifier free epsilon calculus is a strict conservative extension of first order logic.

- Strict: there are formulae not equivalent to any formula of first order logic, e.g. $P(\varepsilon_x Q(x))$ with P, Q unary predicate symbols.
- Conservative: regarding first order formula the epsilon calculus derive the same formulae.



B.6. A false intuition

Although these formulae are related, is not the case that for every formulas S and P one has

$$S(\varepsilon_x P(x)) \dashv\vdash \exists x.(S(x) \wedge P(x))$$

Indeed, let $S(x)$ be $(x = x)$ and let $P(x)$ be $(x \neq x)$ i.e. $\neg S(x)$.

Then

$$S(\varepsilon_x P(x)) \;\equiv\; S(\varepsilon_x.\neg S(x)) \;\equiv\; S(\tau_x.S(x)) \;\equiv\; \forall x.S(x) \;\equiv\; \forall x.x = x$$

which is clearly true.

But $\exists x.(S(x)\&P(x)) \;\equiv\; \exists x.(S(x)\&\neg S(x)) \;\equiv\; \exists x.(x = x \;\&\; x \neq x)$ which is clearly false.

The argument works with any formula of one variable that is universally true like here $S(x) \equiv (x = x)$.

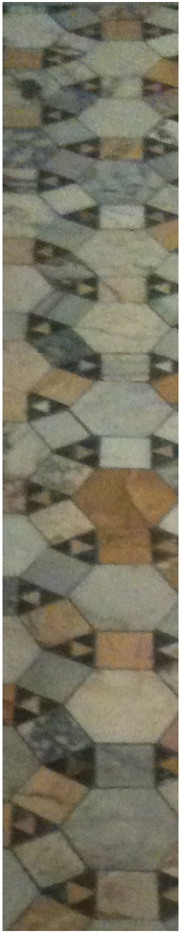
B.7. Main results

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

First epsilon theorem When inferring a formula C without the ε symbol nor quantifiers from formulae Γ not involving the ε symbol nor quantifiers the derivation can be done within quantifier free predicate calculus.

Second epsilon theorem When inferring a formula C without the ε symbol from formulae Γ not involving the ε symbol, the derivation can be done within usual predicate calculus.

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



B.8. Hilbert's view: epsilon substitution method

Epsilon was introduced by Hilbert firstly for arithmetic, in order to establish arithmetic consistency by elementary means. (This was before Gödel's incompletess theorem).

Idea: prove that there is no proof of $0 = 1$ from a finite collection of axioms of Peano arithmetic.

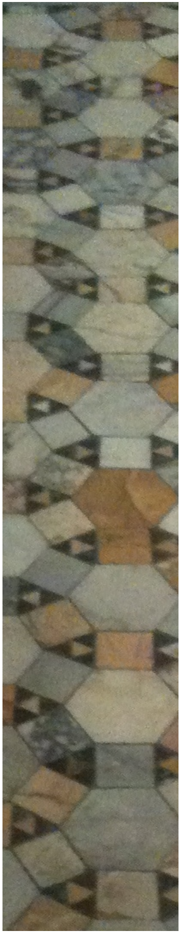
Because of the espilon theorems one may assume that there are no quantifiers rules, only epsilon terms in the instance of the axioms.

Chose a finite set of instance of the axioms.

Find an interpretation of epsilon terms that is "the least integer x such that P(x)" that is $\mu_x P(x)$ that make them all true.

To find them: Start with interpreting all epsilon terms by 0 If the interpretation of some axiom is false, either there is no interpretaiton or the itnerpretation of the epsilon term is too small.

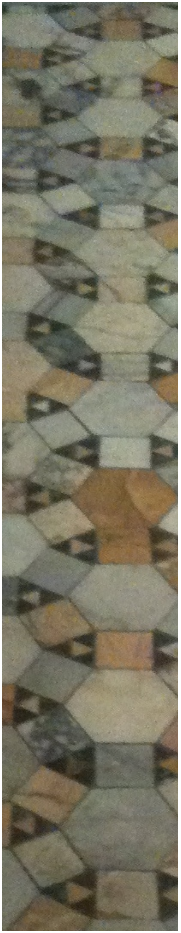
If you can obtain a true assignment of the epsilon terms, as there are no more quantifiers, you can only derive true statements, hence no $0 = 1$.



B.9. Admittedly slightly unpleasant

Heavy notation:

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



B.10. “Loose” use of ε

Some A are B. (E sentences of Aristotle)

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

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

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

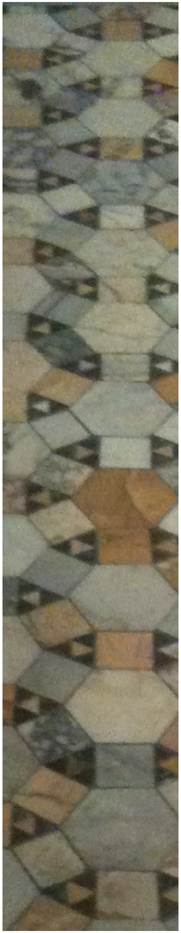
Indeed:

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

On the other hand, one has:

$\exists x. A(x) \& \forall y (A(y) \Rightarrow B(y)) \vdash B(\varepsilon x. A(x))$

because ε-terms are usual terms.



B.11. Intuitive interpretation

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

Here is a pleasant intituitive interpretation rule due to von Heusinger: both "*a*" and "*the*" are interpreted by the an epsilon term, but the "*a*" always refers to a **new** individual in the class, while "*the*" refers to the most **salient** one.

(6) A student entered the lecture hall. He sat down. A student left the lecture hall.

(7) A student arrived lately. The professor looked upset. The student left.

B.12. Categorical model (F. Pasquali)

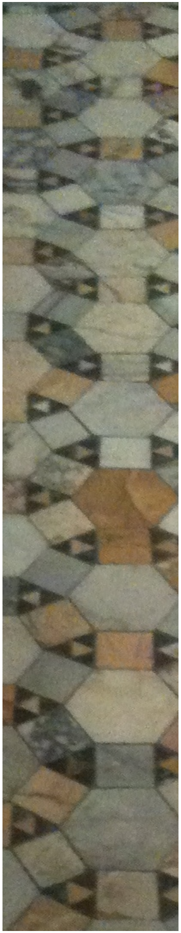
[Sorry for giving little details, this construction is ”heavy” category theory]

Very recently Fabio Pasquali proposed a categorical model: an epsilon logic/language can be interpreted in a Boolean hyper doctrine with a specific property (corresponding to the Axiom of Choice).

Formulae, terms and proofs are all interpreted by arrows. The $\exists x F$ arrow correspond to the ε-term arrow.

Such an hyper doctrine can be constructed from any elementary Topos enjoying the Axiom of choice.

It is important to have a many sorted logic defined with types for his construction.



C Aristotle square of oppositions revisited

C.1. Standard A E I O formulae

- All students passed.

$$A_{P,Q} = \forall x.(P(x) \Rightarrow Q(x))$$

- No student passed.

$$E_{P,Q} = \forall x.(P(x) \Rightarrow \neg Q(x))$$

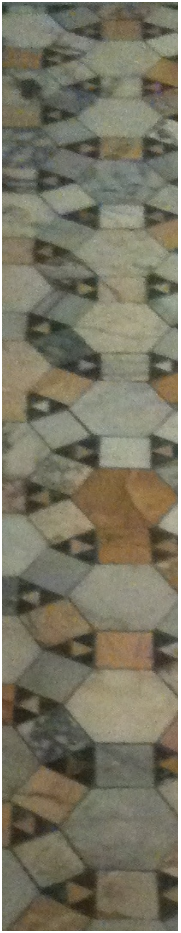
- Some student passed.

$$I_{P,Q} = \exists x.(P(x) \wedge Q(x))$$

- Not all student passed. (original phrasing)

  Some students did not pass. (different focus, but less ambiguous)

$$O_{P,Q} = \exists x.(P(x) \wedge \neg Q(x))$$



C.2. The original square of opposition

Under some conditions (e.g. $\exists x.P(x)$ holds, idealy) the formulae A E I O constitute a square of opposition:

i) $A \dashv\vdash \neg O$ and $E \dashv\vdash \neg I$

ii) It is never the case that $\top \vdash A$ and $\top \vdash E$

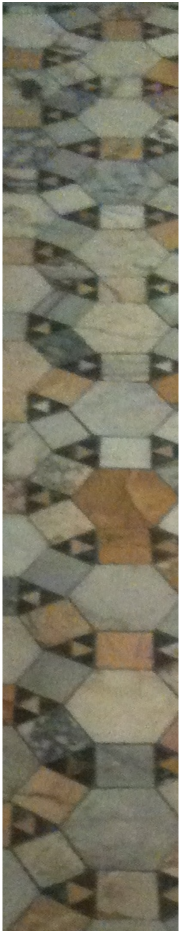
iii) It is never the case that $I \vdash \bot$ and $E \vdash \bot$

iv) $A \vdash I$ and $E \vdash O$

C.3. A picture of the original square

The usual diagrammatical representation is

$$\begin{tikzcd}
A \arrow[rr, leftrightarrow] \arrow[d] \arrow[drr, leftrightarrow] & & E \arrow[d] \arrow[dll, leftrightarrow] \\
I \arrow[rr, leftrightarrow] & & O
\end{tikzcd}$$



C.4. Epsilon E I A O formulae

Consider the epsilon versions of I and A:

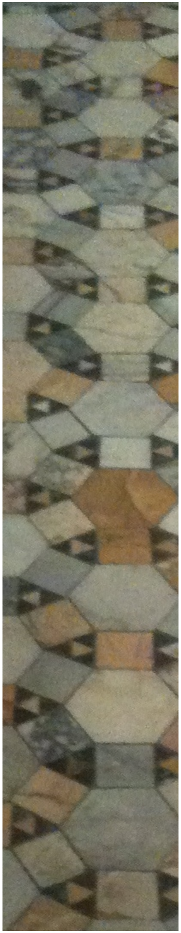
$$I_{S,P} := S(\varepsilon_x P(x))$$

$$A_{S,P} := S(\tau_x P(x))$$

Hence we have no choice for the E and O:

$$E_{S,P} := \neg I_{S,P} = \neg S(\varepsilon_x P(x))$$
$$O_{S,P} := \neg A_{S,P} = \neg S(\tau_x P(x))$$

As we have seen earlier $I_{S,P} \equiv S(\varepsilon_x P(x))$ is not always equivalent to $\exists x. P(x) \& S(x)$



C.5. Hilbertian square of opposition

Let $\mathscr{S}(S,P)$ be the square obtained with the following figures: $A_{S,P}$, $I_{S,P}$, $E_{S,P}$ and $O_{S,P}$.

In the Hilbert's ε-calculus, for every formulas $S(x)$ and $P(x)$ (one free variable), either $\mathscr{S}(S,P)$ or $\mathscr{S}(S,\neg P)$ is a square of opposition. The proof strongly relies on *tertium non datur*. (Pasquali & Retoré)

$$
\begin{tikzcd}
S(\tau_x P(x)) \arrow[r, leftrightarrow] \arrow[d] \arrow[dr, leftrightarrow] & \neg S(\varepsilon_x P(x)) \arrow[d] \arrow[dl, leftrightarrow] \\
S(\varepsilon_P) \arrow[r, leftrightarrow] & \neg S(\tau_P)
\end{tikzcd}
\qquad
\begin{tikzcd}
S(\tau_x \neg P(x)) \arrow[r, leftrightarrow] \arrow[d] \arrow[dr, leftrightarrow] & \neg S(\varepsilon_x \neg P(x)) \arrow[d] \arrow[dl, leftrightarrow] \\
S(\varepsilon_x \neg P(x)) \arrow[r, leftrightarrow] & \neg S(\tau_x \neg P(x))
\end{tikzcd}
$$

D Montagovian computational semantics

D.1. Logic

Logic, philosophy of language, semantics...

Difficult to tell the difference!

From the beginning there are two related parts:

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

There is a link between the two.

In particular quantification if concerned by this link: we quantify over classes which sometimes are implicit, defined by the context.

(8) All students/they prefer going to parties than reading lecture notes.

D.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.

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

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

The language constants for multi sorted First Order Logic:

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

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

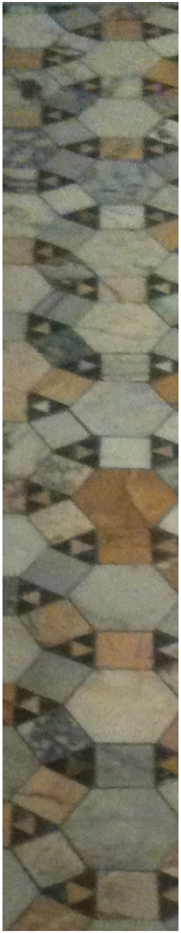
$possibly (\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$

Normal terms (preferably η-long)
type $\mathbf{t}$: **formulae** type $\mathbf{e}$: **terms**.

D.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^*	$=$	$e \rightarrow t$	a noun is a subset of the set of entities
$(A \backslash B)^* = (B/A)^*$	$=$	$A \rightarrow B$	extends easily to all syntactic categories of a Categorial Grammar e.g. a Lambek CG

See lecture by Moot & Retoré next week on the semantics of CG.



D.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”.

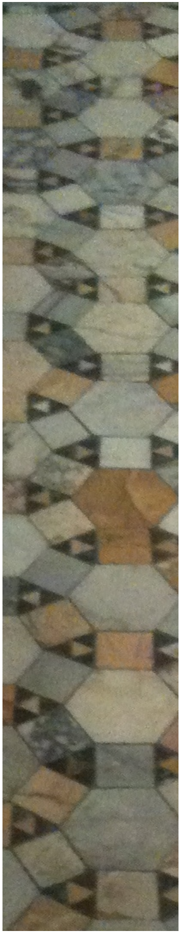
D.7. Ingredients: a parse structure & a lexicon

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



D.8. Computing the semantic representation

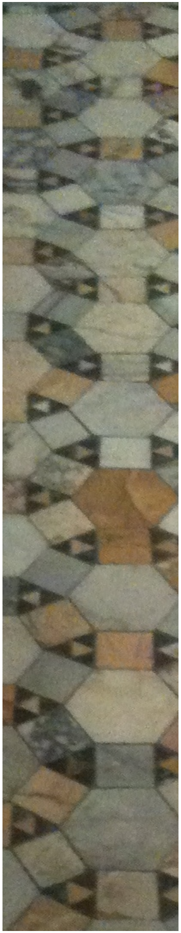
1) Insert the semantics terms into the parse structure

2) β reduce the resulting term

$$\Big(\big(\lambda P^{e\to t}\,\lambda Q^{e\to t}\,(\exists^{(e\to t)\to t}\,(\lambda x^{e}(\wedge(P\;x)(Q\;x))))\big)\big(\lambda x^{e}(\mathtt{club}^{e\to t}\;x)\big)\Big)$$
$$\Big(\big(\lambda y^{e}\,\lambda x^{e}\,((\mathtt{defeated}^{e\to(e\to t)}\;x)y)\big)\;\mathit{Leeds}^{e}\Big)$$
$$\downarrow\beta$$
$$(\lambda Q^{e\to t}\,(\exists^{(e\to t)\to t}\,(\lambda x^{e}(\wedge^{t\to(t\to t)}(\mathtt{club}^{e\to t}\;x)(Q\;x)))))$$
$$(\lambda x^{e}\,((\mathtt{defeated}^{e\to(e\to t)}\;x)\mathit{Leeds}^{e}))$$
$$\downarrow\beta$$
$$(\exists^{(e\to t)\to t}\,(\lambda x^{e}(\wedge(\mathtt{club}^{e\to t}\;x)((\mathtt{defeated}^{e\to(e\to t)}\;x)\mathit{Leeds}^{e}))))$$

Usually human beings prefer to write it like this:

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



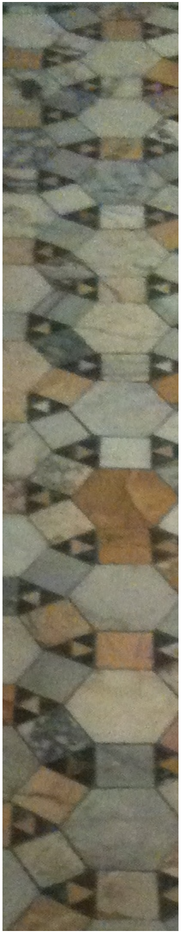
D.9. Montague: good architecture / limits

Good trick (Church):

- a propositional logic for meaning assembly (proofs/λ-terms)
  - **computes**
- HOL / FOL formulae (formulae/meaning; no proofs)

Some limits:

- no account of lexical semantics (restriction of selection, meaning transfers etc.)
- quantification is not as well addressed as it may seem



E Some inadequacies in the Montagovian treatment of quantification

E.1. Usual Montagovian treatment

(1) A tramp died on the pavement.

(2) Something happened to me yesterday.

Usual view (e.g Montague)

Quantifiers apply to the main predicate,

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

and when there is a restriction to a class (e.g. $[\mathit{some}]$) the quantifier applies to two predicates:

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

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

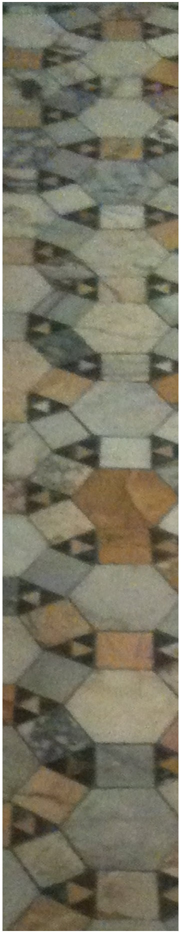
Syntactical structure of the sentence $\neq$ logical form.

(9) Orlando di Lasso composed some motets.

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

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

The underlined predicate is not a proper phrase.



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

Asymmetry class / predicate

(12) a. Some politicians are crooks.
     b. ?? Some crooks are politicians.

(13) a. Some students are employees.
     b. Some employees are students.

The different focus makes a big difference.

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

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

(14) Cars, cars, cars,... (Blog)

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

(16) What a thrill — My thumb instead of an onion. (S. Plath)

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

Even when there is a main predicate, I do think that we interpret the quantified NP as soon as we hear it.

(18) Most students go out on Thursday night.

E.5. Sorts, classes,...

Intuitively, there are several ways to quantify. For instance universal quantification can be viewed:

- as a conjunction over the domain (model theoretical view)
- as a property of the generic member of its class (proof theoretical view)

Completeness makes sure that they both agree.

Nertheless the generic view requires a class, a type.

It is very rare to quantify over all possible entities.

E.6. Sorts and classes for generalised quantifiers

Frege's single sorted logic:

(19) a. $\forall x \in M\ P(x) \quad \equiv \quad \forall x\ (M(x) \Rightarrow P(x))$

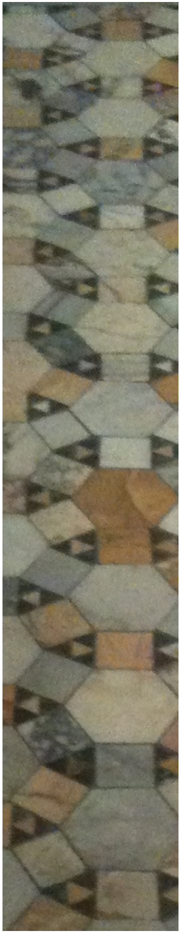
b. $\exists x \in M\ P(x) \quad \equiv \quad \exists x\ (M(x) \& P(x))$

This treatment does not apply to other quantifiers:

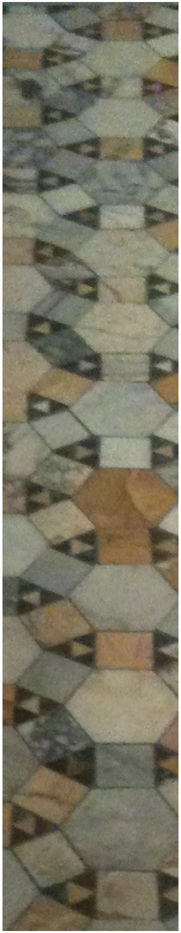
(20) a. for 1/3 of the $x \in M\ P(x) \not\equiv$ for 1/3 of the $x\ (M(x) \Rightarrow P(x))$

b. for few $x \in M\ P(x) \not\equiv$ for few $x\ (M(x) \& P(x))$

Sorts and classes with specific quantifiers may be a good direction.



F The Montagovian generative lexicon ΛTy_n: a many sorted framework



F.1. System F

Types:

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

Terms

- Constants and variables for each type
- $(f^{T \rightarrow U} a^T) : U$
- $(\lambda x^T.\ u^U) : T \rightarrow U$
- $t^{(\Lambda\alpha.\ T)}\{U\} : T[U/\alpha]$
- $\Lambda\alpha.u^T : \Pi\alpha.T$ — 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).

F.2. Basic facts on system F

Logicians / philosophers often ask whether system F is safe?

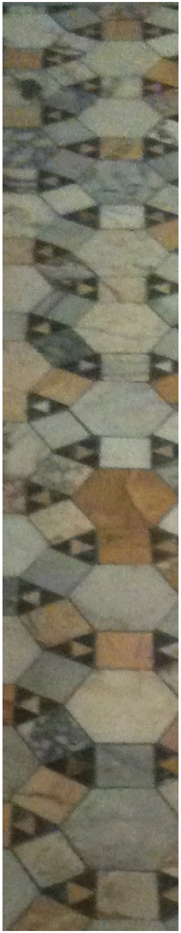
We do not really need system F but any type system with quantification over types. F is syntactically the simplest. (Polynomial Soft Linear Logic of Lafont is enough)

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.)



F.3. Examples of second order usefulness

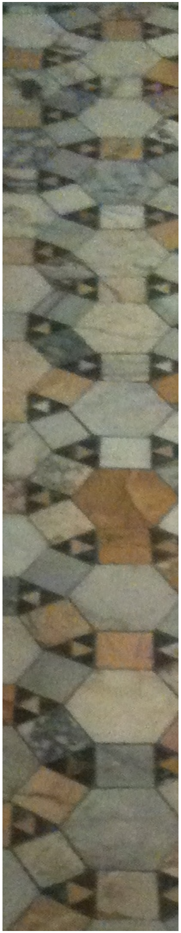
Arbitrary modifiers: $\Lambda\alpha\lambda x^{A}y^{\alpha}f^{\alpha\to R}.((\mathsf{read}^{A\to R\to t}\ x)\ (f\ y))$

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

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



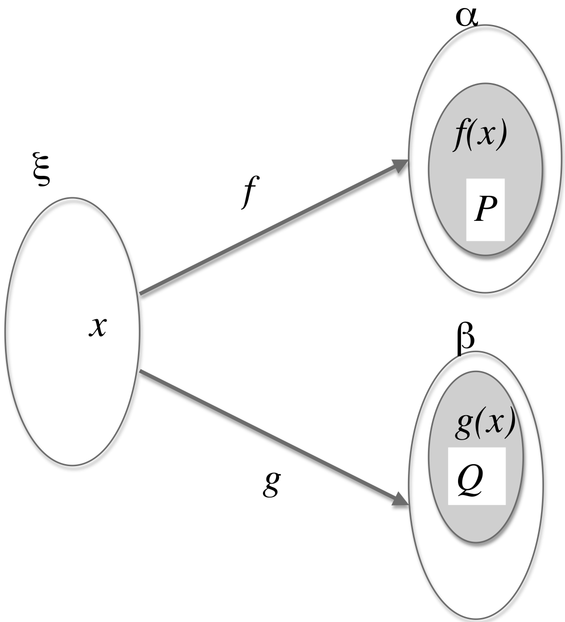


Figure 1: Polymorphic conjunction: $P(f(x))\&Q(g(x))$ with $x : \xi$, $f : \xi \to \alpha$, $g : \xi \to \beta$.

F.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

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

F.5. The Terms: principal or optional

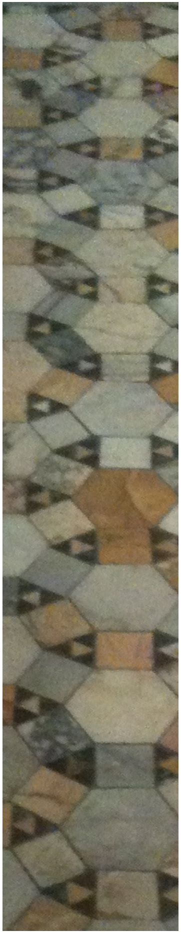
A standard λ-term attached to the main sense:

- 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.



F.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:

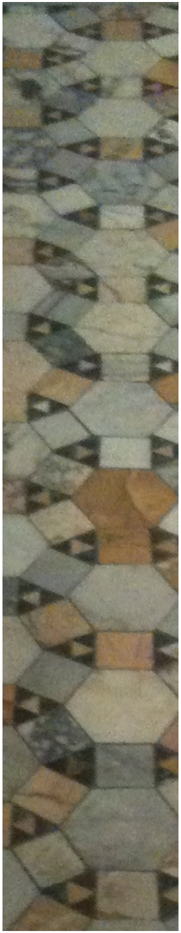
different aspects of the words can be simultaneously used.

F.7. Correct copredication

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

where the base types are defined as follows:

T	Town
F	football club
P	people
Pl	place



F.8. Meaning transfers

(21) Liverpool is spread out.

(22) Liverpool won.

(23) Liverpool voted.

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

Type mismatch, use the appropriate optional term.

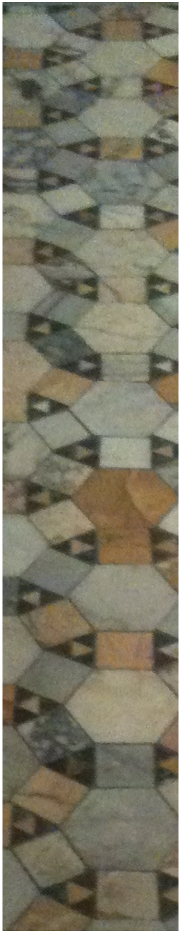
$$spread\_out^{Place \to \mathbf{t}} (t_3^{Town \to Place} Liverpool^{Town})$$

F.9. (In)felicitous copredications

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

(24) Liverpool is spread out and voted.
$\mathit{Town} \rightarrow \mathit{Place}$ and $\mathit{Town} \rightarrow \mathit{People}$
fine

(25) * Liverpool won and voted.
$\mathit{Town} \rightarrow \mathit{FootballClub}$ and $\mathit{Town} \rightarrow \mathit{People}$
blocked because the first transformation is **rigid**.
(sole interpretation: *football* team or committee voted)



F.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.

F.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

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

F.12. Liverpool voted and won

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

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

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*”.

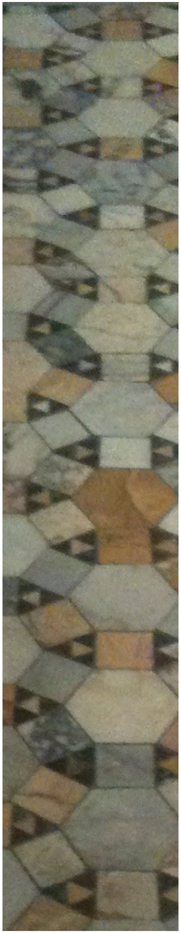
F.13. Other phenomena handled by the same model

Virtual traveller / fictive motion (with Moot & Prévot)
"*The road goes 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.



G Determiners, quantifiers in the Montagovian generative lexicon

G.1. Typed Hilbert operators

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

Many sorted:

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

or

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

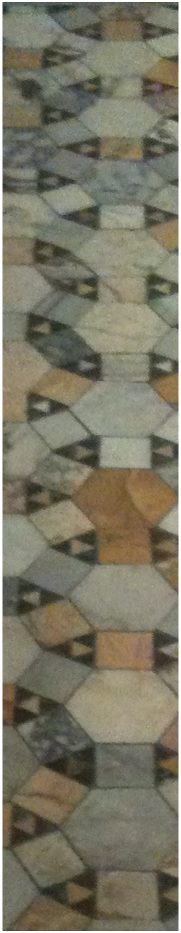
???

either type/formula entails the other:

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

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).



G.2. 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:

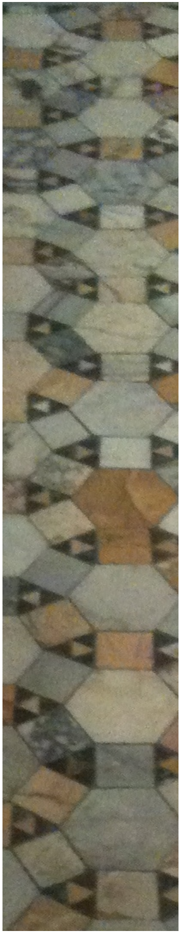
type cat mirrored as a predicate $\widehat{\mathit{cat}} : \mathbf{e} \rightarrow \mathbf{t}$

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.

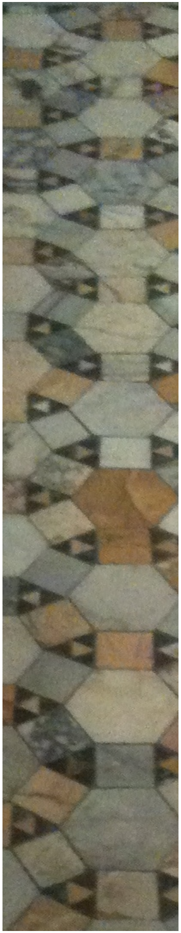


G.3. Computing the proper semantics reading

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

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

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



G.4. Avoiding the infelicities of standard Montague semantics

$\varepsilon_x F$: individual.

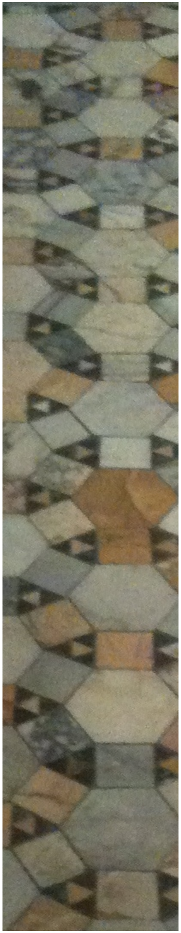
1. Can be interpreted as an individual without the main predicate:
   it is a term.
2. Follows syntactical structure:
   it is a term, the semantics of an NP.
3. Asymmetry subject/predicate:
   $P(\varepsilon Q) \not\equiv Q(\varepsilon P)$.

G.5. E-type pronouns

ε solves the so-called E-type pronouns interpretation (Gareth Evans) where the semantic of the pronoun is the copy of the semantic of its antecedent:

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

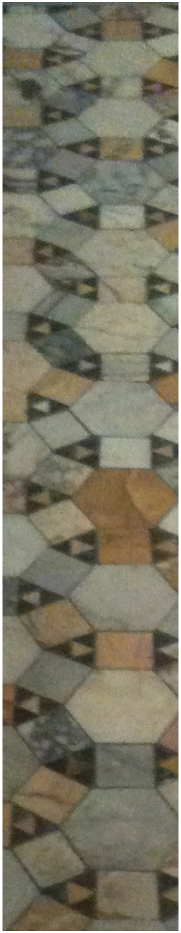
(27) $\text{"}He\text{"} = \text{"}A\ man\text{"} = (\varepsilon_x\ M(x))$.



G.6. 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.
- choice function are not syntactically defined they have to be added one by one in the FOL language.

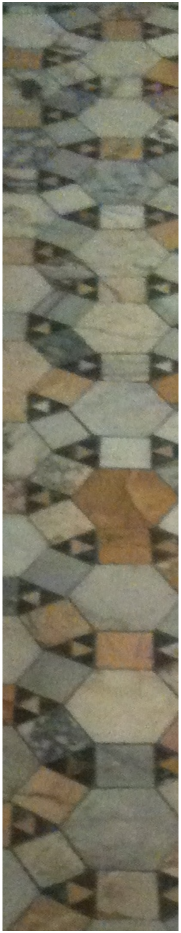


G.7. Universal quantification

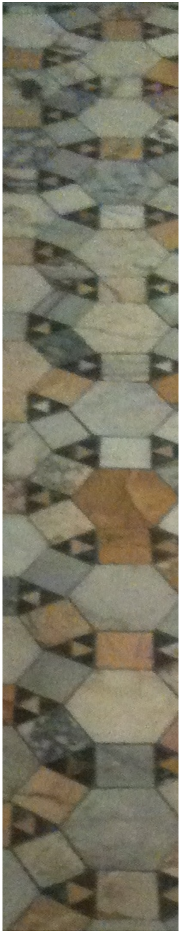
Observe that our setting allow two ways to do so (as for the epsilon):

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

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



H Conclusion

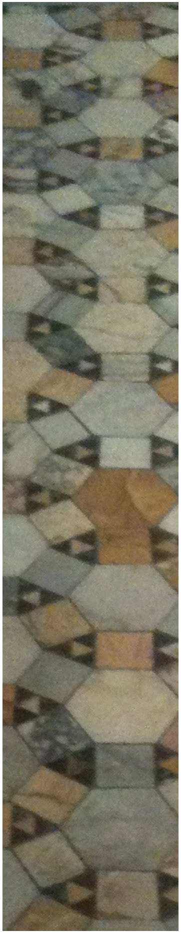


H.1. Semantic analysis of natural language

We propose in our type theoretical framework for lexical and compositional semantics a formulation of quantification, in particular the most frequent ones, existentials, which avoids the drawbacks of the usual interpretation:

- epsilon terms follow the syntactic structure
- they refer to individuals
- they avoid the unpleasant symmetry between CN and VP in existential statements.

The difference and relation between types and properties, type theory and first order logic.



H.2. Logical perspectives

Epsilon open new perspectives on:

- underspecified scope: $P(\tau x.\ A(x), \varepsilon x.\ B(x))$
- generalised quantifiers:

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

- Epsilon in type theory?
- Intuistionistic versions?

H.3. More on Epsilon

HILBERT'S EPSILON AND TAU IN LOGIC, INFORMATICS AND LINGUISTICS

Université de Montpellier 10 11 12 juin 2015

Organised by LIRMM CNRS with the support of ANR Polymnie and Université de Montpellier

https://sites.google.com/site/epsilon2015workshop/