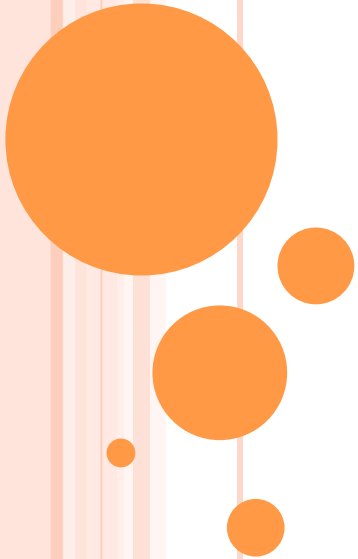


On the Logical modelling of natural quantification

Journées Polymnie Nancy 4-5 février 2016

Christian RETORÉ Université de Montpellier

Quantification is common: examples

- Something happened to me yesterday.
- A man walked into a bar. His work boots were muddy, and he had dirt caked under his fingernails. He wore a grimy hoodie, and kept an arm at his waist, [...]
- There are infinitely many primes.
- Every natural number can be represented as the sum of four integer squares.
- That way we're all writing now.
- All children are artists.

Quantification is common: more examples

- There are several reasons why few students read newspapers in the present.
- Two thirds of the world's inhabitants are clustered in these four regions.
- For the first time, a majority of Californians age 55 and older think that marijuana should be made legal (52% legal, 45% not legal).
- It's fair to say that most Stones fans love Jigsaw.

Quantification is common: more examples

- He wrapped up by explaining the dark future for the Universe when all the stars go away.
- All atoms are made from the same bits, which are called subatomic particles.
- Just about all sentences in the English language fall into ten patterns determined by the presence and functions of nouns, verbs, adjectives, and adverbs.
- All ideas are welcome. (propositions)
- He believes whatever he is being told. (propositions?)

Quantification is common: examples, until overdose.

- In basic math, we're taught multiplication tables. We learn that most numbers are the answer to at least two different multiplication problems, some numbers are the answer to several, and then…
- Any module of known β is weak. Most numbers have even β and most of them are not antisymmetric.
- … thus, in the limit most numbers are not prime.
- **« If all roads lead to Rome, most segments of the transportation system lead to Roma Termini! »**

Automated complete analysis : useful, even in parctice

- The children will have a pizza.
- All children are not afraid of dogs

D.1. Syntaxe catégorielle : principe

catégories S : phrase, np : noun phrase (groupe nominal), n : nom commun

si $w : B/A$ alors w suivi de $u : A$ donne $wu : B$

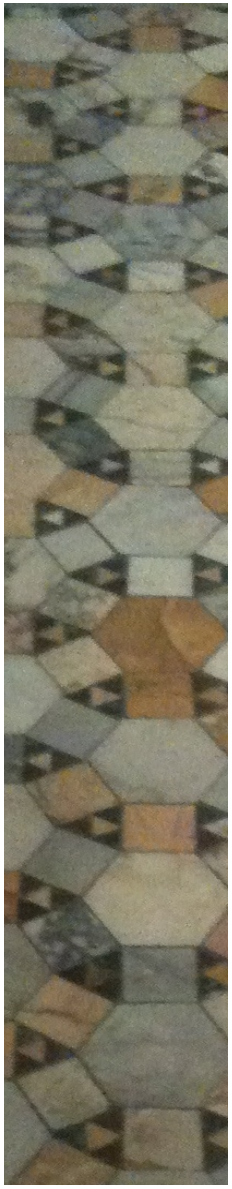
réciproquement, si w suivi de n'importe quoi (une variable) de type A est de type B alors w est de type B/A

au moins deux hypothèses

A hypothèse la plus à gauche

$$\dfrac{\begin{array}{c}\ldots\,[A]\,\ldots\ldots\\ \vdots\\ B\end{array}}{A\backslash B}\ \backslash_i$$ — l'hyp. A est annulée

$$\dfrac{\begin{array}{cc}\Delta & \Gamma\\ \vdots & \vdots\\ A & A\backslash B\end{array}}{B}\ \backslash_e$$



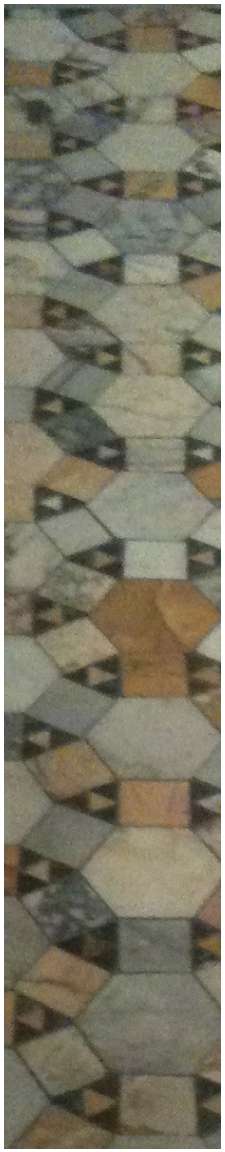
D.2. $A\backslash B$ même chose, mais A est à gauche

au moins deux hyp. libres

A hyp. libre la plus à droite

$$\frac{\begin{array}{c}\ldots\ldots[A]\ldots\\ \vdots\\ B\end{array}}{B/A}\,/_i$$ — l'hyp. A est annulée

$$\frac{\begin{array}{cc}\Gamma & \Delta\\ \vdots & \vdots\\ B/A & A\end{array}}{B}\,/_e$$

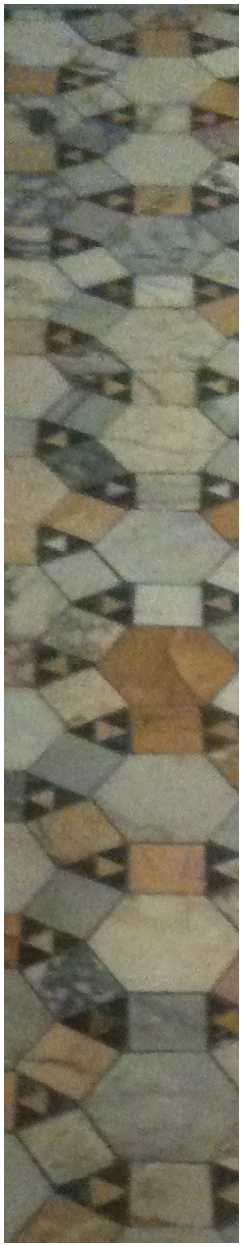


D.3. Analyse syntaxique = déduction (parsing as deduction)

Une phrase est correcte si on peut assigner à chaque mot une catégorie de sorte que la suite des catégories dérive S.

$m_1 \dots m_n$ est une phrase ssi :

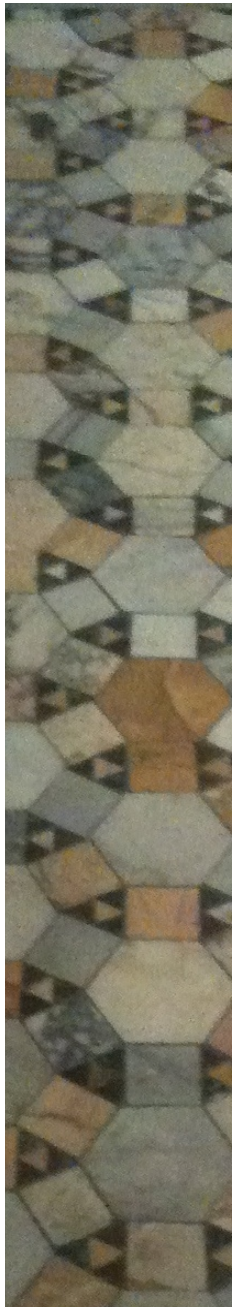
$$\forall i \exists c_i \in Lex(m_i) \quad c_1 \dots c_n \vdash S$$



D.4. Sémantique compositionnelle : principes

Le sens d'une expression composée est fonction du sens de ses parties (Frege) et de leur assemblage syntaxique (Montague)

$(\text{Catégorie syntaxique})^*$	$=$	Type sémantique	
S^*	$=$	t	une phrase est une proposition
np^*	$=$	e	un groupe nominal est une entité/individu
n^*	$=$	$e \to t$	un nom commun est une propriété des entités
$(A\backslash B)^* = (B/A)^*$	$=$	$A \to B$	propage la traduction



D.8. Exemple de calcul sémantique : les enfants prendront une pizza

mot	***catégorie syntaxique*** u ***type sémantique*** u^* ***sémantique :*** λ***-term of type*** u^* x^v ***signifie*** x ***(variable, constante) de type*** v
les	$(S/(np\backslash S))/n$ (subject) $((S/np)\backslash S)/n$ (object) $(e \to t) \to ((e \to t) \to t)$ $\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda x^e(\Rightarrow^{t\to(t\to t)}\ (P\ x)(Q\ x))))$
une	$((S/np)\backslash S)/n$ (object) $(S/(np\backslash S))/n$ (subject) $(e \to t) \to ((e \to t) \to t)$ $\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^e(\wedge^{t\to(t\to t)}(P\ x)(Q\ x))))$
enfant(s)	n $e \to t$ $\lambda x^e(\texttt{enfant}^{e\to t}\ x)$
pizza	n $e \to t$ $\lambda x^e(\texttt{pizza}^{e\to t}\ x)$
prendront	$(np\backslash S)/np$ $e \to (e \to t)$ $\lambda y^e\ \lambda x^e\ ((\texttt{prendront}^{e\to(e\to t)}\ x)y)$

D.9. Analyse syntaxique ∃∀

Il y a deux analyse syntaxique possibles. Une :

∃∀

$$\dfrac{\dfrac{(S/(np\backslash S))/n \quad n}{(S/(np\backslash S))}\,/_e \quad \dfrac{(np\backslash S)/np \quad [np]^1}{(np\backslash S)}\,/_e}{\dfrac{S}{S/np}\,/_i(1) \qquad \dfrac{((S/np)\backslash S)/n \quad n}{(S/np)\backslash S}\,\backslash_e}{S}$$

D.10. Syntaxe $\rightarrow$ λ-terme sémantique de la phrase

$\exists\forall$

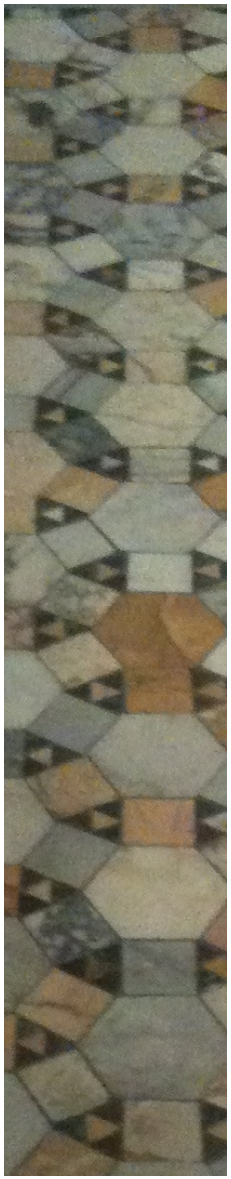
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textit{les}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}} \quad \dfrac{\textit{enfants}}{(\mathbf{e}\rightarrow\mathbf{t})}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}}\rightarrow_e \quad \dfrac{\dfrac{\textit{prendront}}{\mathbf{e}\rightarrow\mathbf{e}\rightarrow\mathbf{t}} \quad \dfrac{\textit{o}}{[\mathbf{e}]^1}}{\mathbf{e}\rightarrow\mathbf{t}}\rightarrow_e}{\mathbf{t}}\rightarrow_e \quad \dfrac{\dfrac{\textit{une}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}} \quad \dfrac{\textit{pizza}}{(\mathbf{e}\rightarrow\mathbf{t})}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}}\rightarrow_e}{\mathbf{e}\rightarrow\mathbf{t}}\rightarrow_i(1) \qquad}{\mathbf{t}}\rightarrow_e$$

Le λ-terme correspondant est :

$\exists\forall = (\textit{une pizza})(\lambda o^{\mathbf{e}}(\textit{les enfants})(\textit{prendront } o))$

Il faut encore :

1. insérer les lambda terme lexicaux et
2. réduire/calculer

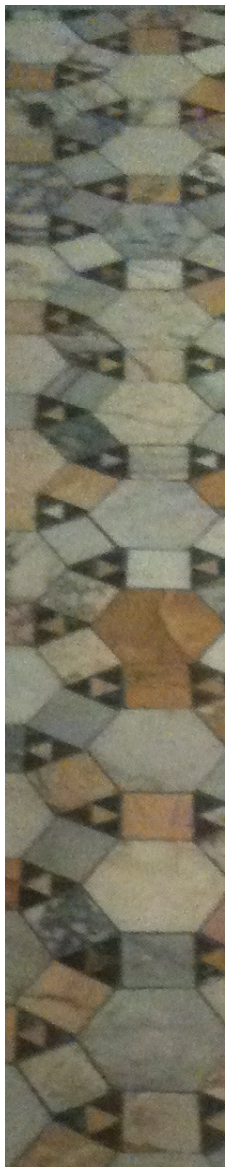


D.11. Calculs, par étapes 1/2

$$
\begin{aligned}
&(\textit{une pizza})\\
&= (\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}(P\ x)(Q\ x)))))(\lambda z^{e}(\texttt{pizza}^{e\to t}\ z))\\
&= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\lambda z^{e}(\texttt{pizza}^{e\to t}\ z))\ x)(Q\ x)))))\\
&= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))(Q\ x))))
\end{aligned}
$$

$$
\begin{aligned}
&(\textit{les enfants})\\
&= (\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda x^{e}(\Rightarrow^{t\to(t\to t)}\ (P\ x)(Q\ x)))))(\lambda u^{e}(\texttt{enfant}^{e\to t}\ u))\\
&= (\lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda x^{e}(\Rightarrow^{t\to(t\to t)}\ ((\lambda u^{e}(\texttt{enfant}^{e\to t}\ u))\ x)(Q\ x)))))\\
&= (\lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda x^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ x)(Q\ x)))))
\end{aligned}
$$

$$
\begin{aligned}
&(\textit{les enfants})(\textit{prendront}\ o) =\\
&(\lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ w)(Q\ w)))))((\lambda y^{e}\ \lambda x^{e}\ ((\texttt{prendront}^{e\to(e\to t)}\ x)\\
&= (\lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ w)(Q\ w)))))(\lambda x^{e}\ ((\texttt{prendront}^{e\to(e\to t)}\ x)\ o))\\
&= \forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ w)((\lambda x^{e}\ ((\texttt{prendront}^{e\to(e\to t)}\ x)\ o))\ w)))\\
&= \forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ w)(((\texttt{prendront}^{e\to(e\to t)}\ w)\ o))))
\end{aligned}
$$

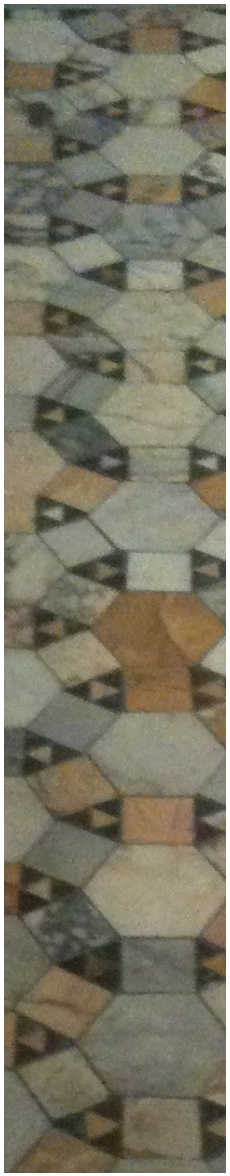


D.12. Calculs, par étapes 2/2

$$
\begin{aligned}
&(\textit{une pizza})(\lambda o\ (\textit{les enfants})(\textit{prendront}\ o))\\
&= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))(Q\ x))))\\
&\qquad (\lambda o \forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ w)(((\texttt{prendront}^{e\to(e\to t)}\ w)\ o)))))\\
&= (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))\\
&\qquad ((\lambda o \forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ w)(((\texttt{prendront}^{e\to(e\to t)}\ w)\ o)))))\ x)))\\
&= (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))\\
&\qquad (\forall^{(e\to t)\to t}\ (\lambda w^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfant}^{e\to t}\ w)((\texttt{prendront}^{e\to(e\to t)}\ w)\ x))))))
\end{aligned}
$$

ce qui s'écrit communément :

$$\exists x.\ \textit{pizza}(x) \wedge \forall w.\ (\textit{enfant}(w) \Rightarrow \textit{prendront}(w,x))$$



D.13. Avec l'autre analyse syntaxique...

$\forall\exists$

$$\dfrac{\dfrac{\dfrac{(S/(np\backslash S))/n \quad n}{(S/(np\backslash S))}\,/_e \qquad \dfrac{\dfrac{\dfrac{[np]^1 \quad \dfrac{(np\backslash S)/np \quad [np]^2}{(np\backslash S)}\,/_e}{S}\,\backslash_e}{S/np}\,/_i(2) \qquad \dfrac{((S/np)\backslash S)/n \quad n}{(S/np)\backslash S}\,\backslash_e}{S}\,\backslash_i(1)}{np\backslash S}\,/_e}{S}$$

Qui correspond à l'analyse :

$\forall\exists$

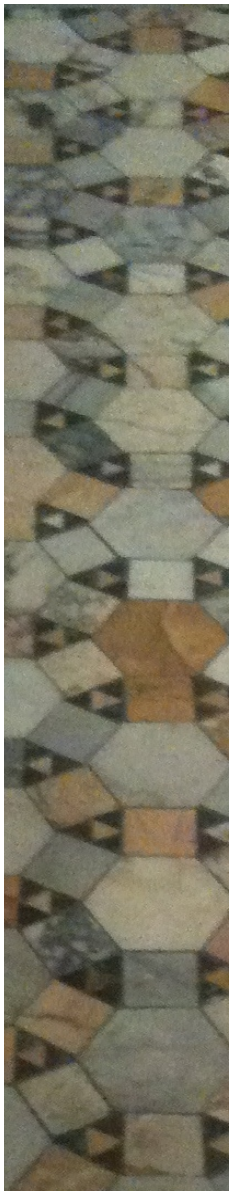
$$\dfrac{\dfrac{\dfrac{\textit{les}\atop (\mathbf{e}\to\mathbf{t})\to(\mathbf{e}\to\mathbf{t})\to\mathbf{t} \qquad \textit{enfants}\atop (\mathbf{e}\to\mathbf{t})}{(\mathbf{e}\to\mathbf{t})\to\mathbf{t}}\to_e}{} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textit{s}\atop [\;\mathbf{e}]^1 \qquad \dfrac{\textit{prendront}\atop \mathbf{e}\to\mathbf{e}\to\mathbf{t} \qquad \textit{o}\atop [\mathbf{e}]^2}{\mathbf{e}\to\mathbf{t}}\to_e}{\mathbf{t}}\to_e}{\mathbf{e}\to\mathbf{t}}\to_i(2) \qquad \dfrac{\textit{une}\atop (\mathbf{e}\to\mathbf{t})\to(\mathbf{e}\to\mathbf{t})\to\mathbf{t} \qquad \textit{pizza}\atop (\mathbf{e}\to\mathbf{t})}{(\mathbf{e}\to\mathbf{t})\to\mathbf{t}}\to_e}{\mathbf{t}}\to_e}{\mathbf{e}\to\mathbf{t}}\to_i(1)}{}}{\mathbf{t}}\to_e$$

λ-terme de la phrase :

$$\forall\exists = (\textit{les enfants})(\lambda s.\ (\textit{une pizza})(\lambda o\ ((\textit{prendront}\ o)\ s)))$$

on insère les λ-termes lexicaux et on calcule

((*une pizza*) et (*les enfants*) déjà faits)



D.14. Calculs (bis repetita placent)

$(\textit{une pizza})(\lambda o\ ((\textit{prendront}\ o)\ s))$
$= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))(Q\ x))))$
$(\lambda o\ (((\lambda y^{e}\ \lambda x^{e}\ ((\texttt{prendront}^{e\to(e\to t)}\ x)\ y))\ o)\ s)))$
$= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))(Q\ x))))$
$(\lambda o\ ((\texttt{prendront}^{e\to(e\to t)}\ s)\ o))$
$= (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))((\lambda o\ ((\texttt{prendront}^{e\to(e\to t)}\ s)\ o))\ x)))$
$= (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))((\texttt{prendront}^{e\to(e\to t)}\ s)\ x)))$

$\forall\exists = (\textit{les enfants})(\lambda s.\ (\textit{une pizza})(\lambda o\ ((\textit{prendront}\ o)\ s)))$
$= (\lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda u^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfants}^{e\to t}\ u)(Q\ u)))))$
$(\lambda s.\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))((\texttt{prendront}^{e\to(e\to t)}\ s)\ x))))$
$= (\forall^{(e\to t)\to t}\ (\lambda u^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfants}^{e\to t}\ u)$
$((\lambda s.\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))((\texttt{prendront}^{e\to(e\to t)}\ s)\ x))))\ u)))))$

$= (\forall^{(e\to t)\to t}\ (\lambda u^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfants}^{e\to t}\ u)$
$(\exists^{(e\to t)\to t}\ (\lambda x^{e}.\ (\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))((\texttt{prendront}^{e\to(e\to t)}\ u)\ x))))))$

ce qui s'écrit communément :

$$\forall u.\ \textit{enfants}(u) \Rightarrow \exists.x\ \textit{pizza}(x) \wedge \textit{prendront}(u,x)$$

Weaknesses of the standard categorial analysis

- At least in Lambek grammars median quantification cannot be handled
- The syntactic structure is cosi-cosi
- One syntactic category per possible position of the quantifier
- What's the interpretation of a quantified NP? (type raising / generalised quantifiers P,Q->prop.)

CHOMSKYAN SYNTAX

- Good synctactic structure
- Then quantfier raising (covert movement)
- Not that different from the catagorial analysis
- Slightly obscure and ad hoc (how does it work when the quantifiers appear in sub propositions
- What is the logical form ? How is it computed?
- What is the interpretation of the quantified NP?

Underspecified semantics (A. Koller, M. Egg, J. Niehren..)

- Semantic representation / formulae that factor the possible readings
- A common representation for the various representations Forall exists / exists forall
  - Competing nodes
  - EXISTS x & …
  - FOR ALL x => …
  - (in this case having sorts or IN would be good.

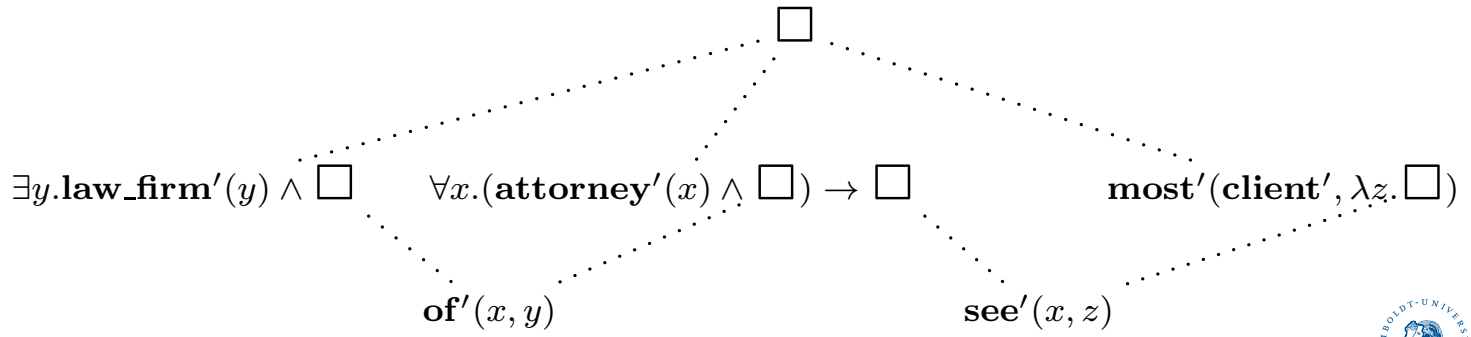
Drawbacks

- Syntax and syntax -> semantics is discussed.
- The streucture of these underspecified formulae (trees is complex)
- I guess using sorts like « Forall a:A » and « Exists a:A » would be slightly more elegant than « forall x A(x) ->…. » and « exists x. A(x) & … »
- Concretely « given an underspecified structure can it lead to a formulae » is an NP complete question

Fields of application and central issues 2: scope

- scope ambiguities are the prototypical structural ambiguities
- scope can be modelled by the relation between fragments and glue points
- to handle scope ambiguity omit the scope relations that make up the differences between the readings
- the set of readings must be constrained appropriately, e.g., for nested NPs

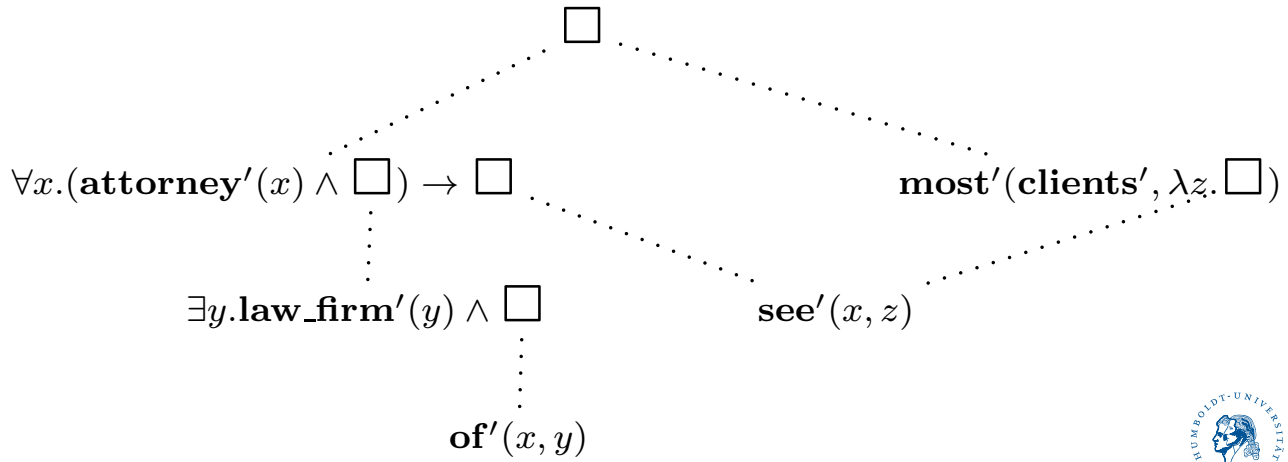
(10) *Every attorney of a law firm saw most clients*



Fields of application and central issues 3: scope

(11) *Every attorney of a law firm saw most clients*

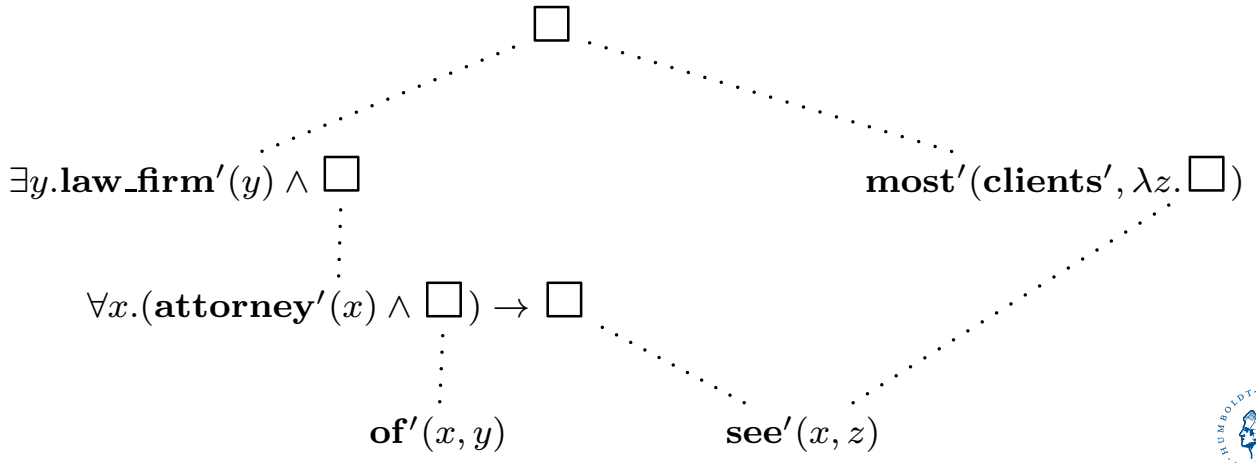
- option 1: give the 'a-law-firm' fragment narrow scope
- then the 'every-attorney' or the 'most-clients' fragment may get widest scope
- i.e., this option describes two readings



Fields of application and central issues 4: scope

(12) *Every attorney of a law firm saw most clients*

- option 2: give the 'a-law-firm' fragment scope over 'every-attorney'
- then the 'most-clients' fragment may scope above, between, or below them
- i.e., this option yields three readings



Categorial minimalist grammars Semantics in Lambda-Mu calculus

- Lambda mu calculus: sort of lambda calculus for classical logic, non confluent reduction (Parigot)
- Use for semantic representations à la Montague (De Groote)
- Maxime Amblard
  - Starting with chomskyan syntax express as deductions (categorial minimalist grammars)
  - Plus lambda DRT to compute the semantics

CMG+LAMBDA-MU DRT (AMBLARD)

- Good syntactic structures
- Underspecified objects are well defined
- Algorithm for parsing+semantics is clear

- Drawbacks:
  - Proofs for the /syntactic system involve labelling
  - Lambda mu DRT is a bit complicated

lexical item	$\lambda\mu$-DRS & semantic type	syntactic category
the	$\lambda Q.\mu\delta.[[d](Q\ d)] \Rightarrow [(\delta\ d)]]$ $(\mathbf{e} \to \mathbf{t}) \to \mathbf{e}$ specific discourse variable: d	k ⊗ d / n
a	$\lambda Q.\mu\gamma.[p](Q\ p) \wedge (\gamma\ p)]$ $(\mathbf{e} \to \mathbf{t}) \to \mathbf{e}$ specific discourse variable: p	(k ⊗ d) / n
children	$\lambda z.(\mathtt{child}\ z)$ $\mathbf{e} \to \mathbf{t}$	n
pizza	$\lambda z.(\mathtt{piz}\ z)$ $\mathbf{e} \to \mathbf{t}$	n
ate	$\lambda x\lambda y\lambda e.\mathtt{eat}(e, x, y)$ $\mathbf{e} \to \mathbf{e} \to \mathbf{v} \to \mathbf{t}$	V / d
modif	$\lambda R\lambda x_2\lambda y\lambda e.$ $R(y, e) \underline{\wedge} \mathrm{Pa}(e, u)$ $(\mathbf{e} \to \mathbf{v} \to \mathbf{t}) \to \mathbf{e} \to \mathbf{e} \to \mathbf{v} \to \mathbf{t}$	(k\(d\v)) /≤ V
infl	$\lambda Q\lambda y_2\lambda e.Q(e)$ $\wedge \mathrm{P}(e) \underline{\wedge} \mathrm{Ag}(e, y_2)$ $(\mathbf{v} \to \mathbf{t}) \to \mathbf{e} \to \mathbf{v} \to \mathbf{t}$	(k \ t) /≤ v
comp	$\lambda Q.[e](Q(e))]$ $(\mathbf{v} \to \mathbf{t}) \to \mathbf{t}$	C / t

Ta start with: the two deteminer phrases.

$$\frac{\vdash \begin{cases} (\varepsilon \mid \mathrm{a} \mid \varepsilon) \\ (\mathsf{k} \otimes \mathsf{d}) \,/\, \mathsf{n} \\ (\mathbf{e} \to \mathbf{t}) \to \mathbf{e} \\ \lambda Q.\mu\gamma.[p \mid Q(p) \wedge \gamma(p)] \end{cases} \qquad \vdash \begin{cases} (\varepsilon \mid \mathrm{pizza} \mid \varepsilon) \\ \mathsf{n} \\ \mathbf{e} \to \mathbf{t} \\ \lambda z.\mathtt{piz}(z) \end{cases}}{\vdash \begin{cases} (\varepsilon \mid \mathrm{a} \mid \mathrm{pizza}) \\ \mathsf{k} \otimes \mathsf{d} \\ \mathbf{e} \\ \mu\gamma. \\ [p \mid \mathtt{piz}(p) \wedge \gamma(p)] \end{cases}} mg$$

$$\frac{\vdash \begin{cases} (\varepsilon \mid \mathrm{the} \mid \varepsilon) \\ (\mathsf{k} \otimes \mathsf{d}) \,/\, \mathsf{n} \\ (\mathbf{e} \to \mathbf{t}) \to \mathbf{e} \\ \lambda Q\mu\delta.[\,|\,[d \mid Q(d)] \Rightarrow [\delta(d)]] \end{cases} \qquad \vdash \begin{cases} (\varepsilon \mid \mathrm{children} \mid \varepsilon) \\ \mathsf{n} \\ \mathbf{e} \to \mathbf{t} \\ \lambda z.\mathtt{child}(z) \end{cases}}{\vdash \begin{cases} (\varepsilon \mid \mathrm{the} \mid \mathrm{children}) \\ \mathsf{k} \otimes \mathsf{d} \\ \mathbf{e} \\ \mu\delta.[\,|\,[d \mid \mathtt{child}(d)] \\ \quad \Rightarrow [\,|\,\delta(d)]] \end{cases}} mg$$

$$\frac{\vdash \begin{cases} (\varepsilon \mid \text{eat} \mid \varepsilon) \\ \mathsf{V} \,/\, \mathsf{d} \\ \mathbf{e} \to \mathbf{e} \to \mathbf{v} \to \mathbf{t} \\ \lambda x \lambda y \lambda e.\text{eat}(e, x, y) \end{cases} \qquad \begin{matrix} u{:}\mathsf{d} \\ \underline{u}{:}\mathbf{e} \end{matrix} \vdash \begin{cases} (\varepsilon \mid u \mid \varepsilon) \\ \mathsf{d} \\ \mathbf{e} \\ \underline{u} \end{cases}}{\begin{matrix} u{:}\mathsf{d} \\ \underline{u}{:}\mathbf{e} \end{matrix} \vdash \begin{cases} (\varepsilon \mid \text{eat} \mid u) \\ \mathsf{V} \\ \mathbf{e} \to \mathbf{v} \to \mathbf{t} \\ \lambda y \lambda e. \\ \text{eat}(e, \underline{u}, y) \end{cases}} mg$$

$$\frac{\vdash \begin{cases} (\varepsilon \mid \varepsilon \mid \varepsilon) \\ (\mathsf{k}\backslash(\mathsf{d}\backslash\mathsf{v})) \mathbin{/_{=}} \mathsf{V} \\ (\mathbf{e}\to\mathbf{v}\to\mathbf{t})\to\mathbf{e}\to\mathbf{e}\to\mathbf{v}\to\mathbf{t} \\ \lambda R\lambda x_2\lambda y\lambda e. \\ R(y,e) \mathbin{\underline{\wedge}} \mathsf{Pa}(e,x_2) \end{cases} \qquad \begin{matrix} u{:}\mathsf{d} \\ \underline{u}{:}\mathbf{e} \end{matrix} \vdash \begin{cases} (\varepsilon \mid \mathsf{eat} \mid u) \\ \mathsf{V} \\ \mathbf{e}\to\mathbf{v}\to\mathbf{t} \\ \lambda y\lambda e. \\ \mathsf{eat}(e,\underline{u},y) \end{cases}}{\begin{matrix} u{:}\mathsf{d} \\ \underline{u}{:}\mathbf{e} \end{matrix} \vdash \begin{cases} (\varepsilon \mid \mathsf{eat} \mid u) \\ \mathsf{k}\backslash(\mathsf{d}\backslash\mathsf{v}) \\ \mathbf{e}\to\mathbf{e}\to\mathbf{v}\to\mathbf{t} \\ \lambda x_2\lambda y\lambda e.\mathsf{eat}(e,\underline{u},y) \\ \mathbin{\underline{\wedge}}\mathsf{Pa}(e,x_2) \end{cases}}\ hdr$$

$$\frac{\begin{array}{ll} v{:}\mathsf{k} & \vdash \begin{cases} (\varepsilon \mid v \mid \varepsilon) \\ \mathsf{k} \\ \mathbf{e} \\ \underline{v} \end{cases} \\ \underline{v}{:}\mathbf{e} & \end{array} \qquad \begin{array}{ll} u{:}\mathsf{d} & \vdash \begin{cases} (\varepsilon \mid \mathrm{eat} \mid u) \\ \mathsf{k}\backslash(\mathsf{d}\backslash\mathsf{v}) \\ \mathbf{e} \to \mathbf{e} \to \mathbf{v} \to \mathbf{t} \\ \lambda x_2 \lambda y \lambda e.\mathrm{eat}(e, \underline{u}, y) \\ \underline{\wedge}\mathrm{Pa}(e, x_2) \end{cases} \\ \underline{u}{:}\mathbf{e} & \end{array}}{\begin{array}{ll} v{:}\mathsf{k}, u{:}\mathsf{d} & \vdash \begin{cases} (v \mid \mathrm{eat} \mid u) \\ \mathsf{d}\backslash\mathsf{v} \\ \mathbf{e} \to \mathbf{v} \to \mathbf{t} \\ \lambda y \lambda e. \\ \mathrm{eat}(e, \underline{u}, y) \\ \underline{\wedge}\mathrm{Pa}(e, \underline{v}) \end{cases} \\ \underline{v}{:}\mathbf{e}, \underline{u}{:}\mathbf{e} & \end{array}}\, mg$$

$$\frac{\vdash \begin{cases} (\varepsilon \mid \mathsf{a} \mid \mathsf{pizza}) \\ \mathsf{k} \otimes \mathsf{d} \\ \mathbf{e} \\ \mu\gamma. \\ [p \mid \mathtt{piz}(p) \wedge \gamma(p)] \end{cases} \qquad v{:}\mathsf{k},\, u{:}\mathsf{d} \qquad \underline{v}{:}\mathbf{e},\, \underline{u}{:}\mathbf{e} \qquad \vdash \begin{cases} (v \mid \mathsf{eat} \mid u) \\ \mathsf{d}\backslash\mathsf{v} \\ \mathbf{e} \to \mathbf{v} \to \mathbf{t} \\ \lambda y \lambda e. \\ \mathtt{eat}(e, \underline{u}, y) \\ \underline{\wedge}\mathrm{Pa}(e, \underline{v}) \end{cases}}{\vdash \begin{cases} (\mathsf{a\,pizza} \mid \mathsf{eat} \mid \varepsilon) \\ \mathsf{d}\backslash\mathsf{v} \\ \mathbf{e} \to \mathbf{v} \to \mathbf{t} \\ \lambda y \lambda e. \\ \mathtt{eat}(e, \mu\gamma.[p \mid \mathtt{piz}(p) \wedge \gamma(p)], y) \\ \underline{\wedge}\mathrm{Pa}(e, p) \end{cases}} mv$$

$$\frac{\begin{array}{l} w{:}\mathbf{d} \\ \underline{w}{:}\mathbf{e} \end{array} \vdash \left\{\begin{array}{l} (\varepsilon \,|\, w \,|\, \varepsilon) \\ \mathbf{d} \\ \mathbf{e} \\ \underline{w} \end{array}\right. \qquad \vdash \left\{\begin{array}{l} (\mathrm{a\,pizza} \,|\, \mathrm{eat} \,|\, \varepsilon) \\ \mathbf{d}\backslash\mathbf{v} \\ \mathbf{e} \to \mathbf{v} \to \mathbf{t} \\ \lambda y \lambda e. \\ \mathtt{eat}(e, \mu\gamma.[p \,|\, \mathtt{piz}(p) \wedge \gamma(p)], y) \\ \underline{\wedge}\mathrm{Pa}(e, p) \end{array}\right.}{\begin{array}{l} w{:}\mathbf{d} \\ \underline{w}{:}\mathbf{e} \end{array} \vdash \left\{\begin{array}{l} (w\ \mathrm{a\,pizza} \,|\, \mathrm{eat} \,|\, \varepsilon) \\ \mathbf{v} \\ \mathbf{v} \to \mathbf{t} \\ \lambda e. \\ \mathtt{eat}(e, \mu\gamma.[p \,|\, \mathtt{piz}(p) \wedge \gamma(p)], \\ \underline{w})\underline{\wedge}\mathrm{Pa}(e, p) \end{array}\right.}\ mg$$

$$\frac{\begin{array}{ll} y{:}k & \vdash \left\{\begin{array}{l}(\varepsilon \,|\, y \,|\, \varepsilon)\\ \mathbf{k}\\ \mathbf{e}\\ \underline{y}\end{array}\right. \qquad w{:}\mathbf{d} \\ \underline{y}{:}\mathbf{e} & \qquad \underline{w}{:}\mathbf{e} \vdash \left\{\begin{array}{l}(\varepsilon \,|\, \mathtt{ate} \,|\, w \text{ a pizza})\\ \mathbf{k} \setminus \mathbf{t}\\ \mathbf{e} \to \mathbf{v} \to \mathbf{t}\\ \lambda y_2 \lambda e.\\ \mathtt{eat}(e, \mu\gamma.[p \,|\, \mathtt{piz}(p) \wedge \gamma(p)], \underline{w})\\ \underline{\wedge}\mathrm{Pa}(e, p) \underline{\wedge} \mathrm{P}(e)\\ \underline{\wedge}\mathrm{Ag}(e, y_2)\end{array}\right. \end{array}}{\begin{array}{ll} w{:}\mathbf{d}, y{:}k & \vdash \left\{\begin{array}{l}(y \,|\, \mathtt{ate} \,|\, w \text{ a pizza})\\ \mathbf{t}\\ \mathbf{v} \to \mathbf{t}\\ \lambda e.\\ \mathtt{eat}(e, \mu\gamma.\\ \quad [p \,|\, \mathtt{piz}(p) \wedge \gamma(p)],\\ \underline{w})\\ \underline{\wedge}\mathrm{Pa}(e, p) \underline{\wedge} \mathrm{P}(e)\\ \underline{\wedge}\mathrm{Ag}(e, \underline{y})\end{array}\right. \\ \underline{w}{:}\mathbf{e}, \underline{y}{:}\mathbf{e} & \end{array}}\, ms$$

$$\frac{\vdash \left\{\begin{array}{l}(\varepsilon \mid \text{the} \mid \text{children}) \\ \mathsf{k} \otimes \mathsf{d} \\ \mathbf{e} \\ \mu\delta.[\,|\,[d \mid \mathtt{child}(d)] \Rightarrow [\,|\,\delta(d)]]\end{array}\right. \quad \begin{array}{l} w{:}\mathsf{d}, \mathsf{y}{:}\mathsf{k} \\ \underline{w}{:}\mathbf{e}, \underline{y}{:}\mathbf{e}\end{array} \quad \vdash \left\{\begin{array}{l}(\mathsf{y} \mid \text{ate} \mid w \text{ a pizza}) \\ \mathbf{t} \\ \mathbf{v} \rightarrow \mathbf{t} \\ \lambda e. \\ \mathtt{eat}(e, \mu\gamma. \\ \quad [p \mid \mathtt{piz}(p) \wedge \gamma(p)], \\ \underline{w}) \\ \underline{\wedge}\mathrm{Pa}(e, p) \underline{\wedge} \mathrm{P}(e) \\ \underline{\wedge}\mathtt{Ag}(e, \underline{y})\end{array}\right.}{\vdash \left\{\begin{array}{l}(\text{the children} \mid \text{ate} \mid \text{a pizza}) \\ \mathbf{t} \\ \mathbf{v} \rightarrow \mathbf{t} \\ \lambda e.\mathtt{eat}(e, \mu\gamma.[p \mid \mathtt{piz}(p) \wedge \gamma(p)], \\ \quad \mu\delta.[\,|\,[d \mid \mathtt{child}(d)] \Rightarrow [\,|\,\delta(d)]]) \\ \underline{\wedge}\mathrm{Pa}(e, p) \underline{\wedge} \mathrm{P}(e) \underline{\wedge} \mathtt{Ag}(e, d)\end{array}\right.} \; mv$$

Finally, the complementiser turns the whole sentence into a complement by *Merge*, and semantically reifies the whole formula.

$$\frac{\vdash \begin{cases} (\varepsilon \,|\, \varepsilon \,|\, \varepsilon) \\ \mathsf{C} \,/\, \mathbf{t} \\ (\mathbf{v} \rightarrow \mathbf{t}) \rightarrow \mathbf{t} \\ \lambda Q.[e \,|\, (Q(e))] \end{cases} \quad \vdash \begin{cases} (\text{the children} \,|\, \text{ate} \,|\, \text{a pizza}) \\ \mathbf{t} \\ \mathbf{v} \rightarrow \mathbf{t} \\ \lambda e.\mathtt{eat}(e, \mu\gamma.[p \,|\, \mathtt{piz}(p) \wedge \gamma(p)], \\ \quad \mu\delta.[\,|\,[d \,|\, \mathtt{child}(d)] \Rightarrow [\,|\,\delta(d)]]) \\ \underline{\wedge}\mathrm{Pa}(e,p) \underline{\wedge} \mathrm{P}(e) \underline{\wedge} \mathtt{Ag}(e,d) \end{cases}}{\vdash \begin{cases} (\varepsilon \,|\, \varepsilon \,|\, \text{the children ate a pizza}) \\ \mathsf{C} \\ \mathbf{t} \\ [e \,|\, \mathtt{eat}(e, \mu\gamma.[p \,|\, \mathtt{piz}(p) \wedge \gamma(p)], \\ \quad \mu\delta.[\,|\,[d \,|\, \mathtt{child}(d)] \Rightarrow [\,|\,\delta(d)]]) \\ \underline{\wedge}\mathrm{Pa}(e,p) \underline{\wedge} \mathrm{P}(e) \underline{\wedge} \mathtt{Ag}(e,d)] \end{cases}} mg$$

Thus the syntactic derivation yields the following $\lambda\mu$-DRS:

$$[e|\texttt{eat}(e, \mu\gamma.[p|\texttt{piz}(p) \wedge \gamma(p)], \mu\delta.[[[d|\texttt{child}(d)] \Rightarrow [|(\delta\ d)]]) \underline{\wedge}\texttt{Pa}(e, p)\underline{\wedge}\texttt{P}(e)\underline{\wedge}\texttt{Ag}(e, d)]$$

$$[e|\texttt{eat}(e,$$
$$\mu\gamma.[p|\texttt{piz}(p) \wedge \gamma(p)\wedge\texttt{Pa}(e, p)],$$
$$\mu\delta.[[[d|\texttt{child}(d)] \Rightarrow [|\delta(d)\wedge\texttt{Ag}(e, d)]])$$
$$\wedge\texttt{P}(e)]$$

If we insert the result of X into the result of W and add the final P(e) we obtain: $[e|[p|(\texttt{piz}\ p) \wedge [|[d|(\texttt{child}\ d)] \Rightarrow$
$[|(((\texttt{eat}\ e)p)d) \wedge ((\texttt{Ag}\ e)d)]] \wedge ((\texttt{Pa}\ e)p)] \wedge \mathrm{P}(e)]$

that is one of the two possible readings:

$$\exists e.\mathrm{P}(e) \wedge$$
$$\exists p\,(\texttt{piz}(p) \wedge \texttt{Pa}(e, p) \wedge$$
$$\forall d\,(\texttt{child}(d) \Rightarrow (\texttt{eat}(e, p, d) \wedge \texttt{Ag}(e, d))))$$

If we insert the reduct of S into what we reduced W to, we get:

$$W \rightsquigarrow$$
$$[[[d](\texttt{child}\, d)]$$
$$\Rightarrow [[[p](\texttt{piz}\, p) \wedge (((\texttt{eat}\, e)p)d) \wedge ((\texttt{Pa}\, e)p)] \wedge ((\texttt{Ag}\, e)d)]]$$

If we add the $[e|\mathrm{P}(e) \wedge \cdots$ that has been left out, we obtain the other possible reading, namely:

$$\exists e\, \mathrm{P}(e) \wedge$$
$$\forall d\, (\texttt{child}(d) \Rightarrow \texttt{Ag}(e, d) \wedge$$
$$\exists p(\texttt{piz}(p) \wedge \texttt{eat}(e, p, d) \wedge \texttt{Pa}(e, p)))$$

Thus we have derived the two readings of "*the children ate a pizza*" from a single semantic representation as a $\lambda\mu$DRS that was derived from the syntactic derivation without raising the type of the quantifiers.

General drawbacks syntax/semantics

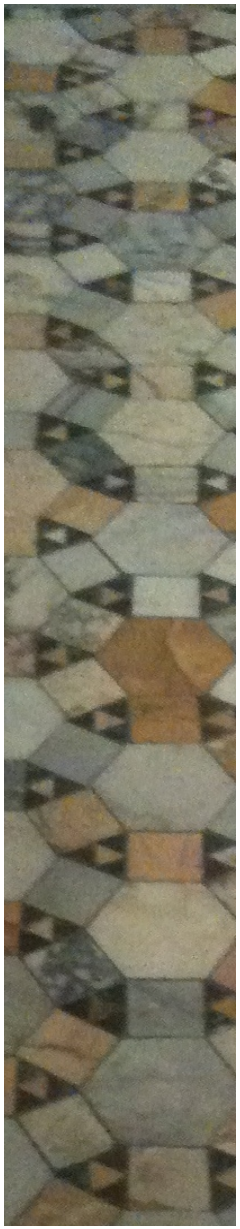
- What is the interpretation of a quantified NP ?
- The interpretation does not follow the syntactic structure (compositionality problem)
- Asymetry of the existentials?
  - Some politicians are crooks
  - ?? Some crooks are politicians
  - (no one is interested in the transverse class of « crooks » it is more of a predicate than of a sort)
  - Some students are employees
  - Some employees are student
  - (different contexts, different meaning)

Choice functions

- Conversion to prenex form (then: non ambiguous)
- Extension of the logical language for the specific reading of a specific sentence Goob point: follows the syntacitc structure
- Quantified NP have an intepretation (as individual terms)

Epsilon and tau (Hilbert)

- Follows the syntactic structure
- Quantified NP are meaningful per se.
- No need to intrduce specific constants for each sentence s in the logical language
- Kind of underspecification
- Difficulty: goes beyond Fol / HOL no models



E.1. Usual Montagovian treatment

(1) A tramp died on the pavement.

(2) Something happened to me yesterday.

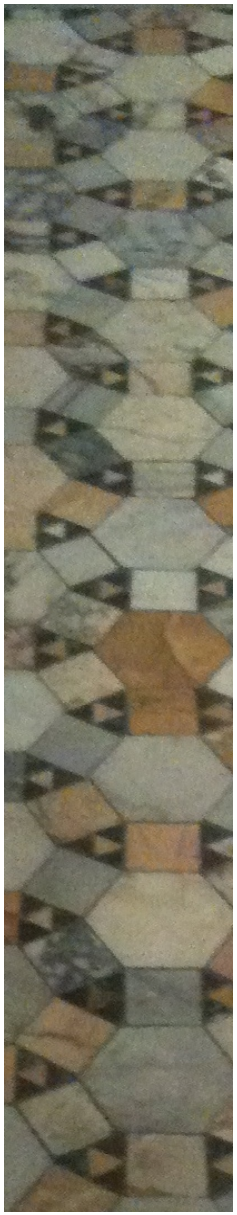
Usual view (e.g Montague)

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)) (λx. OdL composed x)

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)

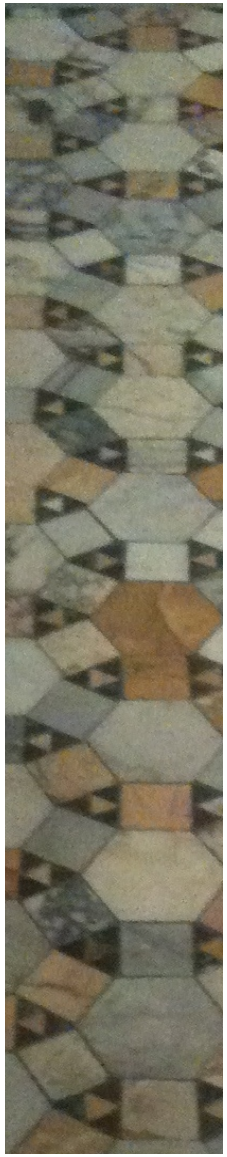
(15) Premier voyage, New-York. (B. Cendrars)

(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.



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 $sleeps(x : cat)$

Type to Formula:

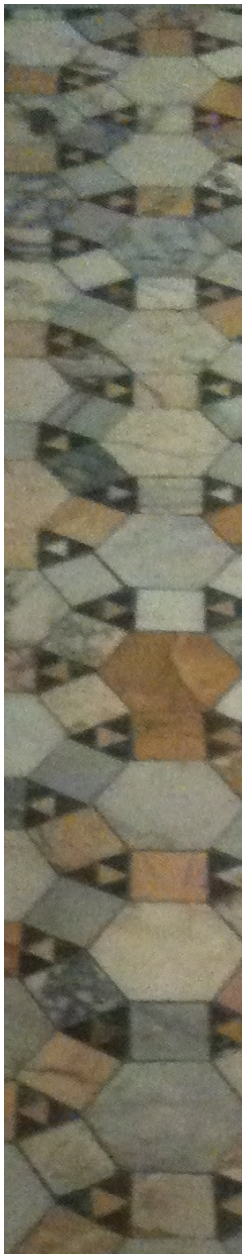
type cat mirrored as a predicate $\widehat{cat} : \mathbf{e} \to \mathbf{t}$

Formula to Type?

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

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

At least it is not a natural class.

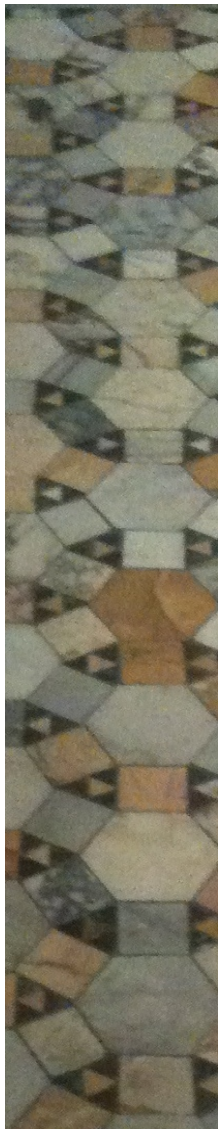


G.3. Computing the proper semantics reading

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

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

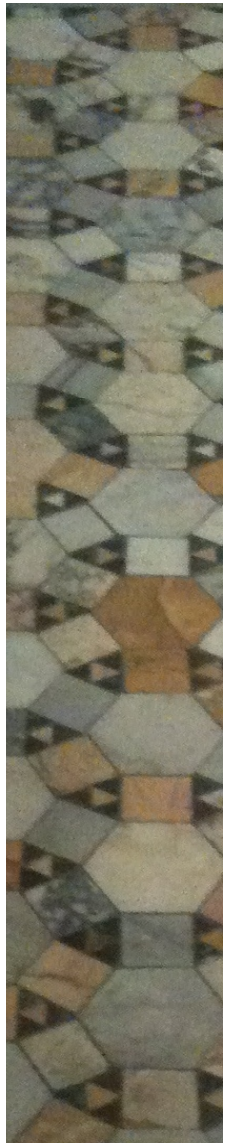
For applying ε to a type say cat,
any type has a predicative counterpart cat (type) $\widehat{cat} : \mathbf{e} \to \mathbf{t}$.
(domains can be restrained / extended)



G.4. Avoiding the infelicities of standard Montague semantics

$\varepsilon_x F(x)$: 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:

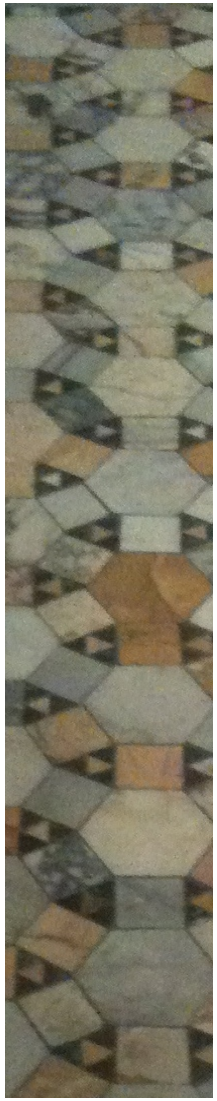
(26) A man came in. He sat dow.

(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 $\Pi\alpha.\ (\alpha \to \mathbf{t}) \to \alpha$

Preferences ???

- When there are several readings one can see that there are preferred readings.
- Studies: syntactic left rigth preference:
  - A circle is connected to every square.
- But common sense reasonning may conflict with this and lead to prefer another reading.
  - A guard stands in front of each gate.
- I do not know any good formal account of preferences of judgements.

INTERPRETATION

- Standard interpretation:
  standard formulae with standard models
  (even with possible worlds)
  « for all/exists x P(x) » is unterpreted as (meta)
  for all/exists x in D « P(x) » holds
- Epsilon formulae difficult to interpret when they are not equivalent to FOL formulae
- How do we distinguish between
  TOUT and CHAQUE
  which have the **same truth conditions**
  but different **usages**
- **Other sorts of interpretations are needed.**

Focus: distinguishing « tout » from « chaque » with assertability conditions

- Joint work with Alda MARI (CNRS IJN-ENS)

"Chaque vin a sa lie" vs. 'Toute nuit a un jour"

Does the difference in the human processing of "chaque" and "tout" match the difference between the proof rules for conjunction and quantification?

Alda Mari
CNRS IJN-ENS Paris
University of Chicago

Christian Retoré
Université de Montpellier
LIRMM CNRS

(IN)COHERENCES 3 – Nancy December 3-4 2015

Generic statements

Tout only can be used in generic sentences.

(4) a. Tout homme est mortel.
TOUT man is mortal.
b. #Chaque homme est mortel.
CHAQUE man is mortal.

→ *Tout* unrestricted generality.

Generic statements

Tout only can be used if no elements of the class. Which s typical of generic sentences.
If no student got an A:

(5) a. Tout étudiant ayant eu un A a un prix.
TOUT student who got an A has a price.
b. #Chaque étudiant qui a eu un A a un prix.
CHAQUE student who got an A has a price.

Restricting the domain

(6) a. Chaque plante de mon jardin est verte.
CHAQUE plant in my garden is green.
b. Toute plante de mon jardin est verte.
TOUT plant in my garden is green.

The b. sentence is about the types of plants that are allowed in my garden, not necessarily about the actual plants which are in my garden.
$\rightarrow$ *Tout* creates a type reading when restricted.

Restricting the domain

(7) a. #Tout homme sur terre est mortel.
TOUT man on earth is mortal. (sounds odd)
b. Chaque homme sur terre est mortel.
CHAQUE man on earth is mortal. (sounds as a weak generalization, but true)

The restriction has the effect of narrowing the domain for *chaque*. This restriction is needed for *chaque* but it not with *tout*. A subtype is created with *tout* and hence the oddness, as the a. sentence suggests that there are types on non earthy men who are not mortal.

Tout and Prescriptivity

Tout requires an underlying rule:
$\Box P(x) \rightarrow Q(x)$.
Tout is used in prescriptive statements (similarly to indefinite generic statements, Cohen, 2001).

(12) a. Tout chien a un systme nerveux.
TOUT dog has a nervous system.

Having a nervous system is part of being a dog.

Chaque and Descriptivity

Chaque requires that one investigates, one by one, all the members of the class.
Recall: *chaque* requires a closed domain of quantification. It conveys that all the members have been inspected.
There is no rule underlying the use of *chaque*, and any property can be used.

Chaque and Descriptivity

We expect:

- Different distributions of the types of properties they can combine with.
- Different patterns of tolerance to exceptions.

Essential vs. accidental properties

Tout is only compatible with essential properties (subtrigging car rescue, but not always).

(13) a. Tout enfant est joyeux.
TOUT child is happy.
b. #Tout enfant est malade.
TOUT child is sick.

Essential vs. accidental properties

Chaque is compatible with both essential and accidental properties. (recall that *chaque* requires that there is a determined domain of quantification).

(14) a. Chaque enfant est joyeux.
CHAQUE child is happy.

b. Chaque enfant est malade.
CHAQUE child is sick.

Tolerance to exceptions

Tout tolerates exceptions as classes

(15) a. Tout enfant est joyeux, sauf les enfants pauvres.
TOUT child is happy, but the poor ones.

Tolerance to exceptions

Tout can tolerate (not very well) individual exceptions.

(16) a. Tout enfant est joyeux, sauf Jean.
TOUT child is happy, but John.

There is an effect though ....

Tolerance to individual exceptions and prescriptivity

Tout can be used in prescriptive statements; it can provide a rule (similarly to indefinite generic statements, Cohen, 2001).
Can it stand individual exceptions ? A first type of individual exceptions.

(17) a. Tout chien a quatre pattes.
TOUT dog has four legs/a brain.
b. Sauf le mien, il a eu un accident.
All but mine, he had an accident.

With the b. sentence you discard the accident information. My dog is a regular dog with 4 legs (it is accidental that he does not have four).

Summarizing ...

Tout

- Compatible with an infinite domain.
- Requires the existence of a law (hence compatible with absence of instances)
- Only compatible with essential properties
- In discourse: it is used prescriptively.

Chaque

- Compatible with both essential and accidental properties
- It requires a well determined domain of quantification (hence incompatible with absence of instances and infinite domains).
- In discourse: it is used descriptively.

Quantification before Frege

Strangely enough first logic is not propositional logic (Stoics) but (restricted) quantified formulae:

A *All* A *are* B

E *Some* A *are* B

I *No* A *is* B

O *Not all* A *are* B / *some* A *are not* B.

Principles before Frege

Rules, patterns (axioms schemes, syllogisms) but no models.

Identity: All A are A
Non contradiction NOT (A and NOT A)

> *Avicenna: Every person refuting the principle of non contradiction should be beaten and burnt until he admits that being beaten is not the same has not being beaten and that being being burnt is not the same has not being burnt.*

Excluded middle : A ou NON A (tertium non datur)

Principles before Frege

Rules, patterns (axioms schemes, syllogisms) but no models.
Syllogisms, e.g. bArOcO

ASSUME All A are B	*A*
ASSUME Not all C are B	*O*
THEREFORE Not all C are A.	*O*

Why quantified sentences firstly? one cannot check that a property holds for each number/triangle (one must forget the figure and reason on the idea of a generic triangle)

British Algebraic Logic XIX: Boole, De Morgan, Pierce, ...

Boole:

$$\forall x.(I(x) \Rightarrow F(x) \vee M(x))$$
$$\Rightarrow (\forall x.(I(x) \Rightarrow F(x)) \vee (\forall x.(I(x) \Rightarrow M(x))))$$

???

Frege: proofs

Proof system: Begriffsschrift
Proof rules (admittedly obscure)
Proofs are finite, tree-like.
Idea of mechanised reasoning (finite and partly computable).
Unique sort: $\forall x{:}A.\ B(x) \equiv \forall x.\ A(x) \Rightarrow B(x)$

$symmetrically\ \exists x{:}A.\ B(x) \equiv \forall x.\ A(x)\&B(x))$

$what\ about\ "most"?$

$\text{most}\, x{:}A.\ B(x) \not\equiv \text{most}\, x.\ A(x) \Rightarrow B(x)$

Frege: Sinn / Bedeutung

The proofs (there can be several non equivalent proofs) of a formula can be seen as its sense (Sinn) cf. e.g. Dummet.

Models , compositional/inductive interpretation of a (logical) sentence, this is commonly viewed as the denotation (Bedeutung) of the sentence. Cf. later.

Hilbert proof systems

A proof is a finite tree starting from axioms, and yielding via a finite set of rules (patterns) to the conclusion.

Deduction theorem: $A \vdash B$ iff $A \Rightarrow B$.

Proof systems : formalisation of mathematical proofs in order to obtain consistency of arithmetics or of analysis etc. by combinatorial arguments on the proofs: if there would be a proof of a contradiction, then every thing would be provable, and there would exists a *normal* proof of $0 = 1$, but there cannot be such a proof. (this method fails for proper theories including arithmetic: cf. Gödel incompleteness theorem)

Hilbert, Gentzen: rules for quantification

Hilbert's rules for quantification:

- $(\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow \forall x \psi)$ (no free x in ϕ)
- $\forall x(\phi \Rightarrow \psi) \Rightarrow ((\forall x \phi) \Rightarrow (\forall x \psi)))$
- $(\forall x \phi) \Rightarrow \phi[x := t]$

Sequents: $X, Y, Z \vdash C$ conclusion C under assumptions X, Y, Z

First rule better stated with sequents:

$$\frac{\Gamma \vdash A(x)}{\Gamma \vdash \forall x.\ A(x)}\ \textit{no free } x \textit{ in } \Gamma$$

Hilbert's τ operator, rules

For any formula $F[x]$ there is a term $\tau_x.F[x]$ (and a term $\epsilon_x F[x] = \tau_x \neg F[x]$)
Rules:

$$\frac{\Gamma \vdash A[x]}{\Gamma \vdash A[\tau_x A[x]]} \textit{ no free } x \textit{ in } \Gamma \qquad \frac{\Gamma \vdash A[\tau_x A[x]]}{\Gamma \vdash A[t]}$$

Hilbert's τ calculus: properties

$\tau_x.F$ enjoys the property F iff everything enjoys F:
$F[\tau_x.F[x]] \equiv \forall x.\ F[x]$

$\epsilon_x.F$ enjoys the property F iff something enjoys F:
$F[\epsilon_x.F[x]] \equiv \exists x.\ F[x]$

Overbinding, in situ quantification: a term inside a predicate of a large formula may have scope over the whole formula.

More formulae than usual: $P(\tau_x Q(x))$ is not equivalent to any usual formula (first or higher order)

First proofs of quantifier elimination and of Herbrand theorem.

Remark: epsilon $\neq$ choice function:
the language (constants, functions, predicates,...) is not extended
this binder is enough for for all formulae at once.

Models (due to Frege, then Löwenheim, Skolem, Gödel)

Model: (family of) situations: set of individuals and interpretation of the constants (individuals, functions, predicates)

Nothing is finite nor computable: even checking the truth of a given formula in a given model can be an infinite process.

As for the propositional calculus the interpretation is flat:

$\forall x P(x)$ is true in M if for all x in M the interpretation of $P(x)$ is true.

Denotation (Bedeutung) of a formula in a model (truth value) , in a family of models (the models in which it is true), etc.

Completeness (Gödel, 1929)

Completeness (Gödel, 1929)
However there is a link between the two:

- F is provable if and only if it is true in any model.
- F can be proved from T if and only if any model that satisfies T satisfies F as well.

Observe that completeness is quite particular for first order logic (classical, and modal intuitionistic with Kripke models)
It fails for second order logic unless one use not-so-natural Henkin models, where predicate vary among definable subsets.

Halfway models/proofs Gentzen ω-rule

Known domain e.g integers for arithmetic which appears as constants in the logical language:

$$\frac{\begin{array}{cccc}\vdots\,\delta(0) & \vdots\,\delta(1) & \vdots\,\delta(2) & \vdots\,\delta(3)\\ \Gamma\vdash A(0) & \Gamma\vdash A(1) & \Gamma\vdash A(2) & \Gamma\vdash A(3)\quad\cdots\end{array}}{\Gamma\vdash\forall x.\ A(x)}\,\omega$$

Very different from the standard rule:

- infinite proof although every branch is finite
- the profos $\delta(i)$ are not necessarily uniforms
- no finite description of the proof (unless there is a description of the proof $\delta(n)$ from the previous $\delta(i)$ with $i \leq n$).
- $\forall x.\ A(x)$ looks like $\&_{i:integer} A(i)$ but this is not a first order formula.

Gentzen ω-rule versus standard rule

$\forall \geq \omega$ Standard rule $\rightarrow$ ω-rule Observe that if one has a proof $\delta(x)$ with a generic element x (a variable not free in any hypothesis) of $P(x)$ ie. when the classical rules works, one can have an omega version (provided the integers are constants of the language) by specialising $\delta(x)$ to each/every number to get $\delta(0)$, $\delta(1)$, $\delta(2)$, $\delta(3)$, ...

$\forall \not\leq \omega$ ω-rule $\not\rightarrow$ standard rule The converse does not hold, unless all the $\delta(i)$ are uniform, have the same shape and do not make use of any particularity of i.

A way to express the ω rule is to assert that $\forall \leq \omega$.

As we want to distinguish the two, we may write $\mathrm{chaque}(x : D)\ A(x)$, instead of $\forall x.A(x)$ as the conclusion of the ω-rule, that is a mere shorthand for $\&_{i:D}A(i)$ this presupposes that the basis of the (possible) world(s) is known.

TOUT: standard rule (generalisation)

When can we correctly assert a TOUT sentence?
As said above TOUT matches rather well the standard rule.
Indeed, TOUT has to be established by reasoning:

- its domain can be a sort, an infinite collection or not so well defined collection, that cannot be throughly examined
- it has a sempiternal nature, it is a rule

What about exceptions: even in maths we intend to make such mistakes and to correct them afterwards:

> *for instance we can wrongly derive* $\forall n.1/n \leq 1$ *when* n *is an integer, one often forgets that* n *cannot be* 0 *and then fix it afterwards:* $\forall n.n \neq 0 \Rightarrow 1/n \leq 1$

> *this works as well (or even better, since we needed to refer to an element) when the exception corresponds to a property:* $\forall n.\ 1/(n \mathit{mod} 2) = 1$ *is fixed as* $\forall n.\ \mathit{Odd}(n) \Rightarrow 1/(n \mathit{mod} 2) = 1$

Chaque: models or ω inspired rule

$\mathrm{chaque}(x : D)\ A(x)$ can be asserted when the domain is known and when for any x in D one has $A(x)$ hence it is a mere shorthand for $\&_{i:D} A(i)$ this presupposes that the basis of the (possible) world(s) is known.
Indeed, CHAQUE rather correspond to a thorough inspection of every element in the domain of quantification,$\&_{i \in D} A(i)$ (which is not a first order formula) [The model approach corresponding to can be supported, but it is a different framework.]

Now if we think at the situation in which one can assert CHAQUE it is because we have a proof or evidence for every entity x in the domain that $A(x)$ holds, (hence the form of the rule is similar to the one of the omega rule).

Comparing the assertion conditions of TOUT and CHAQUE

One often uses CHAQUE while TOUT can be asserted. This is fairly normal: if one is able to say TOUT, if there is a rule, than the conjunction for a precise domain at a precise moment can be deduced from the generic proof, by specialisation, as said above about proofs.

Refutation of universal quantification

In order to test this correspondence,
how do we refute CHAQUE and TOUT
in our opinion:

- Refutation of CHAQUE : find a counter example and that's all! The asserter needs to accept or redefine the domain
- Refutation of TOUT show that a subclass A (we remain with properties and generic associated with these properties) and add this as a restriction so one obtains the provable formula $\forall x.A(x) \Rightarrow P(x)$ — the counter example like the previous condition $n \neq 0$ is a particular case, that's a class with one element.

Forhtcoming experiments

2 groups of students, spring 2016
Web questionaries (with limited time per question)

- situation with precise and imprecise domains, with finite and infinite domains,
- preferred way to express a situation
- true or not in a situation
- preferred refutations of a given quantified sentence

DataBase postgresql/php to stock the information on the subjects and the results, and do statistics

Ideally we'd like to test with particular subjects, eg. dyslexic children as some experiments by Delfitto and Vender had interesting results on negation processing with the A E I O statements.

Epilog: “Chaque vin a sa lie.” vs. “Toute nuit a un jour.”

Same structure, same verb but one includes a possessive related to the singular quantifier.

First observe that it is a matter of *preference* and not a yes/no answer. For instance, when swapping the two quantifiers the two resulting variants of the proverbs sound not that bad. Also observe that “sa” goes well with “chaque”, and less well with “tout”

Nevertheless, before experiments are made, an intuitive analysis at those two proverbs supports our claim:

- the wine is much more concrete, and one can think of each of them as the content of a barrel, because each of them as its lie (which lies in the barrel).
- night/day are even more metaphorical and abstract, more infinite, they seem to be essences that are constant in time.