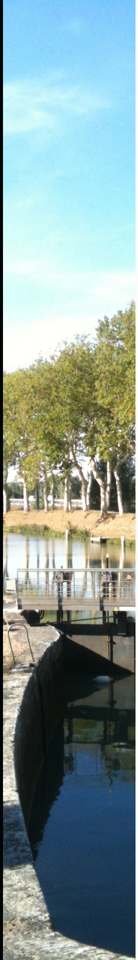
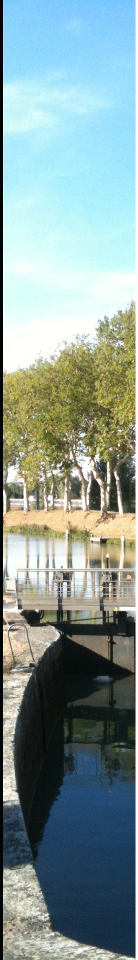


A type theoretical framework for compositional semantics and lexical pragmatics — determiners and quantifiers —

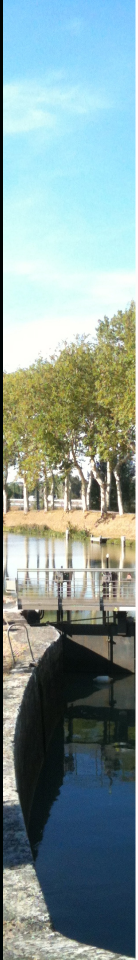
Christian Retoré

Université de Bordeaux & IRIT (en 2012–2013)

Workshop on the logic of the lexicon



A Comparison with previous talks



A.1. Resemblances (in particular with Asher, Luo)

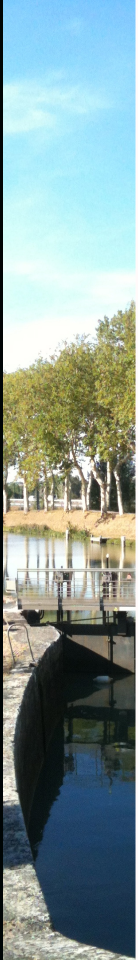
Integration into compositional semantics of lexical aspects and not the other way round.

Type mismatch to block impossible readings

Rich type systems: common names as base types (what about nary predicates like verbs? For instance, is eat of type $\mathit{animate} \rightarrow \mathit{object} \rightarrow \mathbf{t}$ or is there a base type eat with e.g. $\mathit{devour} : \mathit{eat}$?)

Words themselves provides a very good abstract logical language.

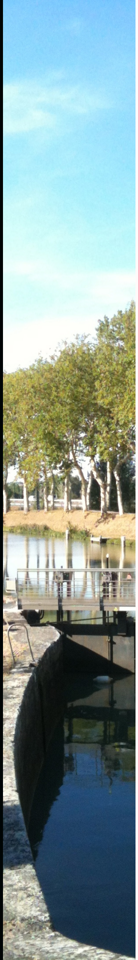
Morphisms to implement meaning transfers



A.2. Common architecture

Two logics, with some convergence between the two.

- Metalogic, logic for meaning assembly (proofs, terms): simply typed λ-calculus with $\mathbf{e}$, $\mathbf{t}$ (Montague) Ty_n (Muskens)
- Logic in which semantic representations are formulated (FOL, higher order, multi sorted,..)



A.3. A common principle: predicate and argument trigger meaning transfers

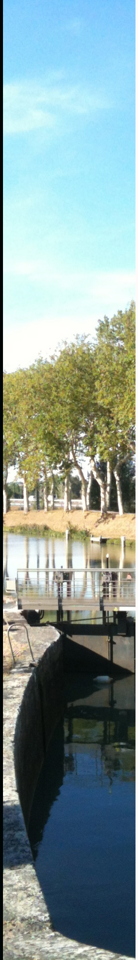
Both argument and predicate can contribute to meaning transfer (cf. Nunberg)

(1) Read the subway wall this afternoon as I made my way to the office.

(2) I knew a lady very well in the village where I grew up, who used to read coffee.

books are informational objects and informational object are likeable.

There is a symmetry between predicates and arguments (= CN?) : observe they both are properties, of type $(\mathbf{e} \rightarrow \mathbf{t})$ in Montague.



A.4. Epsilonian differences (with Asher, Luo)

Extra expressive power of system F (filtered out by syntax).

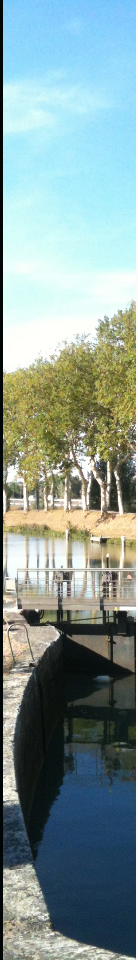
Missing constructs can probalbly **encoded** e.g. $CN = \coprod_{i \in I \text{finite}} cn_i$ with $A \coprod B = \Lambda X.(A \to X) \to (B \to X) \to X$

MTT: predefined types and terms with specific rules

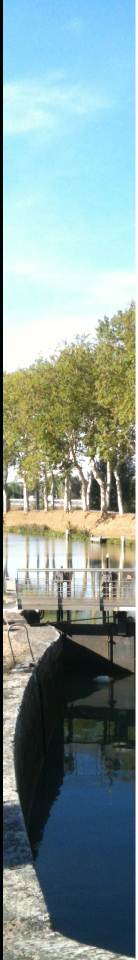
F encoding (unnatural) but can be enriched with terms and operators (e.g. $Y : \Lambda X.\, (X \to X) \to X$ for recursion).

A bigger difference: word trigger meaning transfers.

And **epsilons**, of course!....(although they did not address this issue epsilon surely can be integrated into Asher, Luo work)

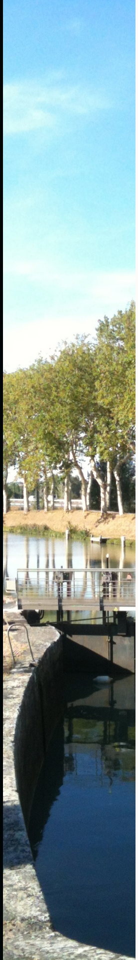


B Second order lambda calculus (Girard's system F)



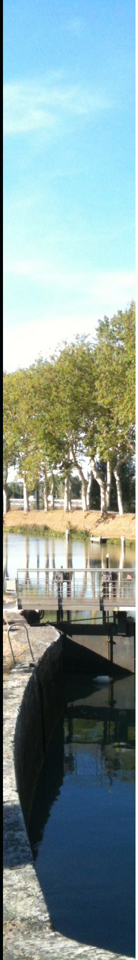
B.1. Types

- Constants types $\mathbf{e}_i$ and $\mathbf{t}$, as well as any type variable $\alpha, \beta, \ldots$ in P, are types.
- Whenever T is a type and α a type variable which may but need not occur in T, $\Lambda\alpha.\ T$ is a type.
- Whenever T_1 and T_2 are types, $T_1 \rightarrow T_2$ is also a type.



B.2. Terms

- A variable of type T i.e. $x : T$ or x^T is a term.
  Countably many variables of each type.
- $(f\ \tau)$ is a term of type U whenever $\tau : T$ and $f : T \to U$.
- $\lambda x^T.\ \tau$ is a term of type $T \to U$ whenever $x : T$, and $\tau : U$.
- $\tau\{U\}$ is a term of type $T[U/\alpha]$ whenever $\tau : \Lambda\alpha.\ T$, and U is a type.
- $\Lambda\alpha.\tau$ is a term of type $\Lambda\alpha.T$ whenever α is a type variable, and $\tau : T$ without any free occurrence of the type variable α. (Type of x in $\forall\alpha.x^\alpha$???)

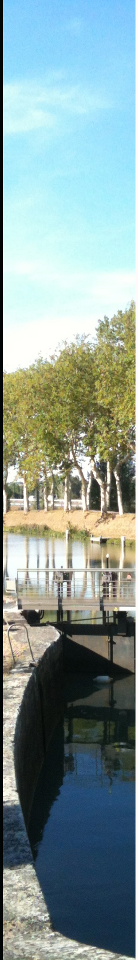


B.3. Extension Curry-Howard

Types are quantified propositional formulae.

Terms are proofs in intuitionistic quantified propositional calculus.

Restriction $q \to \alpha$ and q certainly yield α, but fortunately from this one cannot conclude that under assumption q one has $\forall\alpha.\ \alpha$.



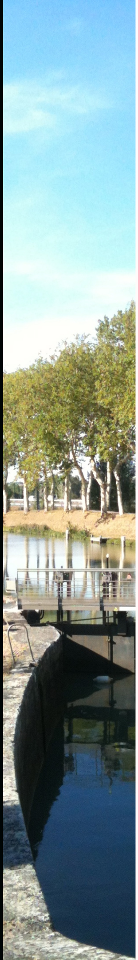
B.4. Reduction

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).

Reduction is strongly normalising and confluent (Girard, 1971): *every term of every type admits a unique normal form which is reached no matter how one proceeds.*

(Difficult: the proof quantifies over all subsets of terms that behaves like stronly normalising terms)



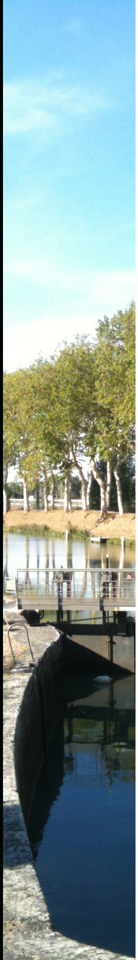
B.5. Unnecessary type operators

The following defined types have the same elimination and introduction behaviour.

Product $A \wedge B$ can be defined as
$\Pi\alpha.\ (A \rightarrow B \rightarrow \alpha) \rightarrow \alpha$

Sum $A \sum B$ can be defined as
$\Pi\alpha.\ (A \rightarrow \alpha) \rightarrow (B \rightarrow \alpha) \rightarrow \alpha$

Existential quantification:
$\Sigma\beta.\ ((\Pi\alpha.\ (V[X] \rightarrow \beta)) \rightarrow \beta)$



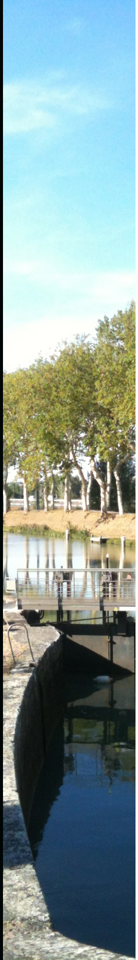
B.6. Inductive types (cf. ML, CaML, Haskell)

Integers
$\Pi\alpha.\ \alpha \to (\alpha \to \alpha) \to \alpha$

List of β objects. (the β can be quantified as well)
$\Pi\alpha.\ \alpha \to (\beta \to \alpha \to \alpha) \to \alpha$

Binary trees
$\Pi\alpha.\ \alpha \to (\alpha \to \alpha \to \alpha) \to \alpha$

Binary trees with leaves L and nodes N
$\Pi\alpha.\ (L \to \alpha) \to (N \to \alpha \to \alpha \to \alpha) \to \alpha$

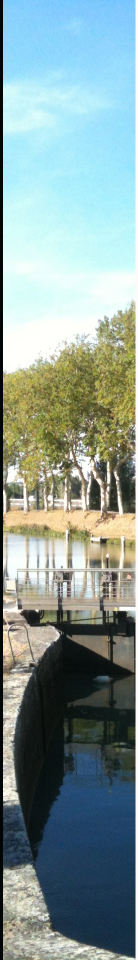


B.7. What can be defined, computed

The function that can be programmed are the ones that can be proved total in second order Heyting arithmetic, that for such issues as the same power as second order Peano arithmetic.

All data types can be defined, and for such types their only normal terms are the expected ones.

More than polymorphic typed functional languages but every program terminates — there is no fixed point operator $Y : \Pi\alpha.\,(\alpha \to \alpha) \to \alpha$.



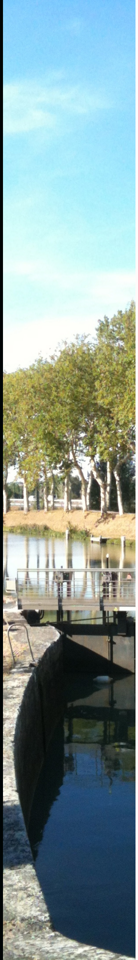
B.8. Coherence, categorical interpretation and normalisation

System F may seem unsafe: one can define a type via reference to all types, including itself. Indeed, some proposed extension collapse.

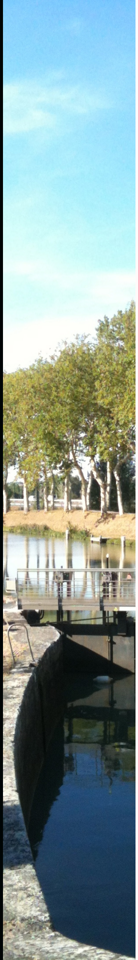
An argument is strong normalisation since there is no normal proof of $\perp = \Pi\alpha.\ \alpha$ (relies on the comprehension axiom for all formulae of HA_2).

Another argument is a concrete categorical interpretation (e.g. with coherence spaces) that shows that there are distinct functions from A to B.

Every term reduces anyhow to its unique normal form. Terms of type $\mathbf{t}$ with constants of mutisorted FOL (resp. HOL) correspond to multisorted formulae of FOL (resp. HOL)



C System F based semantics and pragmatics



C.1. Examples

Dinner was delicious but took ages. (event / food)

* The salmon we had for lunch was lightning fast. (animal / food)

I forgot on the table my preferred book on logic. (physical / info)

I carried the books from the shelf to the attic since i already read them. (phys. / info)

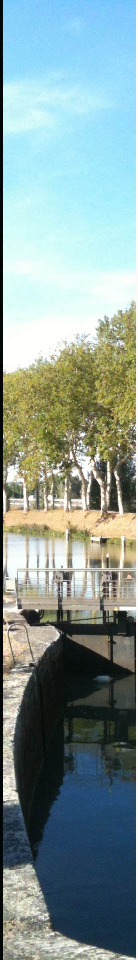
Liverpool is a poor town and an important harbour. (people / geographic)

* Liverpool defeated Chelsea and is an important harbour. (football / geographic)

Nevertheless:

Barcelona won four champions leagues and organised the olympiads.

Libourne, a small south-west town, defeated Lille.



C.2. Types and terms: system F

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

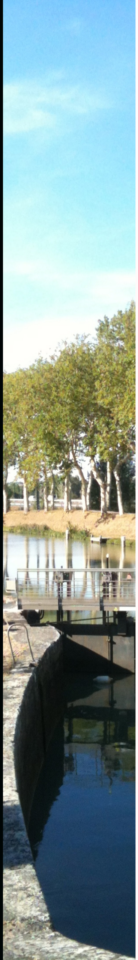
$\mathbf{v}$ (for events who play a particular role in λ-DRT)

$\mathbf{t}$ truth values

types variables roman upper case, greek lower case

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

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



C.3. Examples of second order usefulness

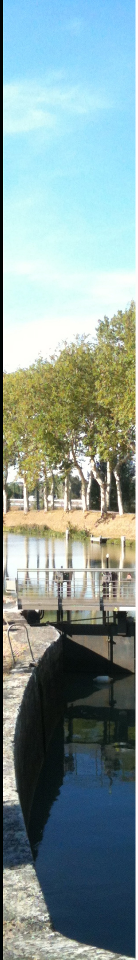
Arbitrary modifiers: $\Lambda\alpha\lambda x^{A} y^{\alpha} f^{\alpha\to R}.((\mathrm{read}^{A\to R\to t}\ x)\ (f\ y))$

Given typesα, β and γ
three predicates $P^{\alpha\to\mathbf{t}}$, $Q^{\beta\to\mathbf{t}}$, $R^{\gamma\to\mathbf{t}}$,
over entities of respective kinds α, β and γ
for any ξ with three morphisms from ξ to α, to β, and to γ

we can coordinate the properties P, Q, R of (the three images of) an entity of type ξ:

AND3= $\Lambda\alpha\Lambda\beta\Lambda\gamma$

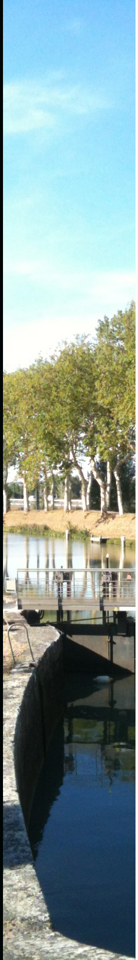
$$\lambda P^{\alpha\to t}\lambda Q^{\beta\to t}\lambda R^{\gamma\to t}$$
$$\Lambda\xi\lambda x^{\xi}$$
$$\lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}\lambda h^{\xi\to\gamma}.$$
$$(\mathrm{and}(\mathrm{and}\ (P\ (f\ x))(Q\ (g\ x)))(R\ (h\ x)))$$



C.4. Principles of our lexicon

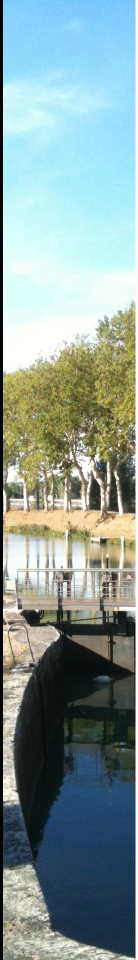
- Remain within realm of Montagovian compositional semantics (for compositionality)
- Allow both predicate and argument to contribute lexical information to the compound.
- Integrate within existing discourse models (λ-DRT).

We advocate a system based on optional modifiers.



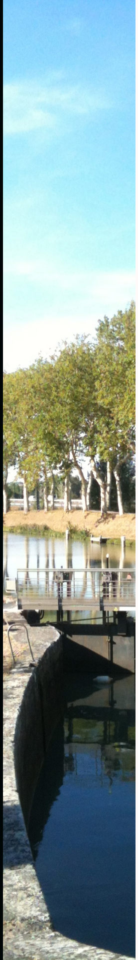
C.5. The Terms: main / standard term

- A standard λ-term attached to the main sense:
  - Used for compositional purposes
  - Comprising detailed typing information
  - Including slots for optional modifiers
  - e.g. $\Lambda\alpha\Lambda\beta\lambda x^{\alpha} y^{\beta} f^{\alpha\to A} g^{\beta\to F}.((\mathsf{eat}^{A\to F\to t}\ (f\ x))\ (g\ y))$
  - e.g. Paris^{T}



C.6. The Terms: Optional Morphisms

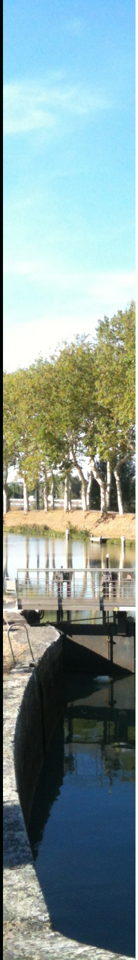
- Each a one-place predicate
- Used, or not, for adaptation purposes
- Each associated with a constraint : rigid, $\varnothing$
  - $\left(\frac{Id^{F \to F}}{\varnothing}, \frac{f_{grind}^{Living \to F}}{rigid} \right)$
  - $\left(\frac{Id^{T \to T}}{\varnothing}, \frac{f_L^{T \to L}}{\varnothing}, \frac{f_P^{T \to P}}{\varnothing}, \frac{f_G^{T \to G}}{rigid} \right)$



C.7. A Complete Lexical Entry

Every lexeme is associated to an *n*-uple such as:

$$\left(\mathsf{Paris}^T, \frac{\lambda x^T.\ x^T}{\varnothing}, \frac{\lambda x^T.(f_L^{T\to L}\ x)}{\varnothing}, \frac{\lambda x^T.(f_P^{T\to P}\ x)}{\varnothing}, \frac{\lambda x^T.(f_G^{T\to G}\ x)}{rigid}\right)$$



C.8. RIGID vs flexible use of optional morphisms

Type clash: $(\lambda x^V.\ (P^{V\to W} x))\tau^U$

$$(\lambda x^V.\ (P^{V\to W} x))\ (f^{U\to V} \tau^U)$$

f: optional term associated with either P or τ

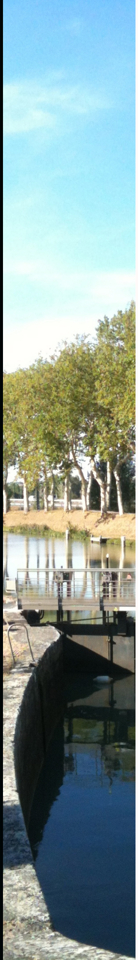
f **applies once to the argument** and not to the several occurrences of x in the function.

A conjunction yields

$(\lambda x^V.\ (\wedge\ (P^{V\to W} x)\ (Q^{V\to W} x))\ (f^{U\to V} \tau^U)$,

the argument is uniformly transformed.

Second order is not needed, the type V of the argument is known and it is always the same for every occurrence of x.



C.9. FLEXIBLEvs.rigid use of optionalmorphisms

$(\lambda x^{?}.\ (\cdots (P^{A\to X} x^{?}) \cdots (Q^{B\to Y} x^{?}) \cdots)\tau^{U}$:
type clash(es) [Montague: $? = A = B$ e.g. e]

$$(\Lambda\xi.\lambda f^{\xi\to A}.\lambda g^{\xi\to B}.\ (\cdots (P^{A\to X}(f x^{\xi})) \cdots (Q^{B\to Y}(g x^{\xi})) \cdots))$$
$$\{U\}\ f^{U\to A}\ g^{U\to B}\ \tau^{U}$$

f, g: optional terms associated with either P or τ.

For each occurrence of x

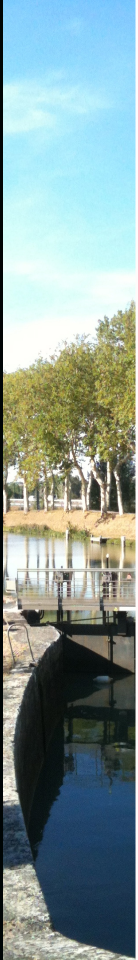
with different A, B, ... with different f, g, ... each time.

Second order typing:

1) anticipates the yet unknown type of the argument

2) factorizes the different function types in the slots.

The types $\{U\}$ and the associated morphism f are inferred from the original formula $(\lambda x^{V}.\ (P^{V\to W} x))\tau^{U}$.



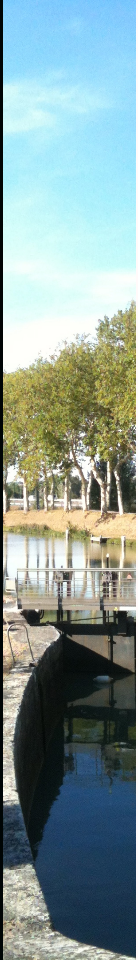
C.10. Standard behaviour

ϕ: physical objects

small stone

$$\overbrace{(\lambda x^{\varphi}.\ (\mathsf{small}^{\varphi\rightarrow\varphi}\, x))}^{\text{small}}\ \overbrace{\tau^{\varphi}}^{\text{stone}}$$

$$(\mathsf{small}\ \tau)^{\varphi}$$



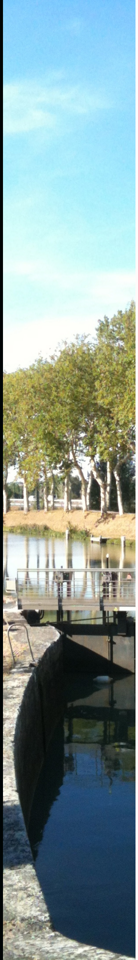
C.11. Qualia exploitation

wondering, loving smile

$$\overbrace{(\lambda x^P.\ (\mathrm{and}^{t\to(t\to t)}\ (\mathrm{wondering}^{P\to t}\ x)\ (\mathrm{loving}^{P\to t}\ x)))}^{\text{wondering, loving}}\ \ \overbrace{\tau^S}^{\text{smile}}$$

$$(\lambda x^P.\ (\mathrm{and}^{t\to(t\to t)}\ (\mathrm{wondering}^{P\to t}\ x)\ (\mathrm{loving}^{P\to t}\ x)))(f_a^{S\to P}\tau^S)$$

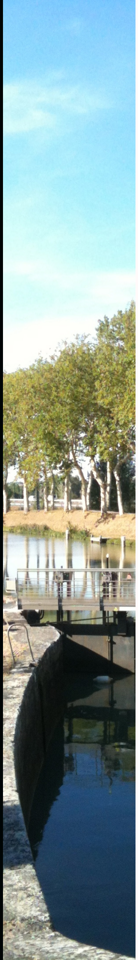
$$(\mathrm{and}\ (\mathrm{loving}\ (f_a\ \tau))\ (\mathrm{loving}\ (f_a\ \tau)))$$



C.12. Facets (dot-objects): incorrect copredication

Incorrect co-predication. The rigid constraint blocks the copredication e.g. $f_g^{Fs \to Fd}$ cannot be **rigidly** used in

(??) The tuna we had yesterday was lightning fast and delicious.

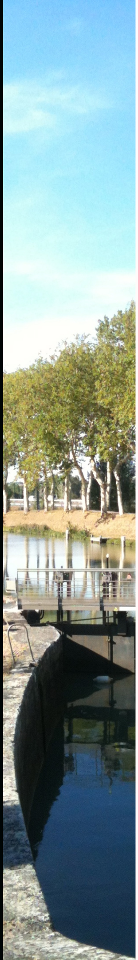


C.13. Facets, correct co-predication. Town example 1/3

T town L location P people

$f_p^{T \to P}$ $f_l^{T \to L}$ k^T København

København is both a seaport and a cosmopolitan capital.



C.14. Facets, correct co-predication. Town example 2/3

Conjunction of $cospl^{P\to t}$, $cap^{T\to t}$ and $port^{L\to t}$, on k^{T}

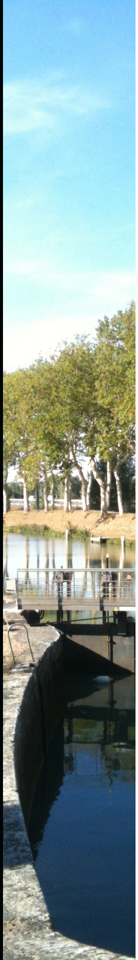
If $T = P = L = e$, (as in Montague)

$(\lambda x^{e}(\mathrm{and}^{t\to(t\to t)}((\mathrm{and}^{t\to(t\to t)}\ (\mathrm{cospl}\ x)\ (\mathrm{cap}\ x))\ (\mathrm{port}\ x)))\ k.$

Conjunction between three predicates... use **AND3**

$$
\begin{array}{l}
\Lambda\alpha\Lambda\beta\Lambda\gamma \\
\qquad \lambda P^{\alpha\to t}\lambda Q^{\beta\to t}\lambda R^{\gamma\to t} \\
\qquad\qquad \Lambda\xi\lambda x^{\xi} \\
\qquad\qquad\qquad \lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}\lambda h^{\xi\to\gamma}. \\
\qquad\qquad\qquad\qquad (\mathrm{and}(\mathrm{and}\ (P\ (f\ x))(Q\ (g\ x)))(R\ (h\ x)))
\end{array}
$$

f, g and h convert x to **different** types (flexible).



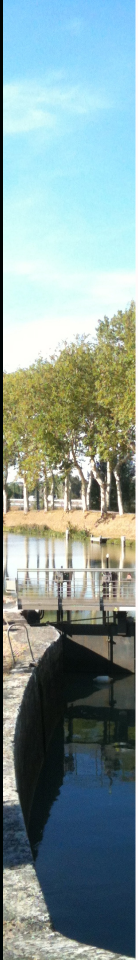
C.15. Facets, correct co-predication. Town example 3/3

AND applied to P and T and L and to $cospl^{P\rightarrow t}$ and $cap^{T\rightarrow t}$ and $\mathrm{port}^{L\rightarrow t}$ yields:

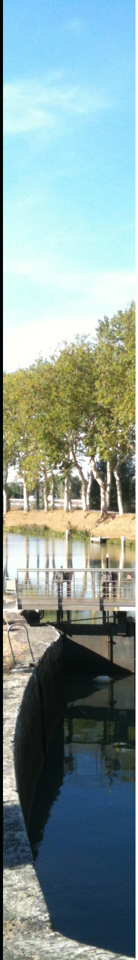
$$\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\rightarrow\alpha}\lambda g^{\xi\rightarrow\beta}\lambda h^{\xi\rightarrow\gamma}.$$
$$(\mathrm{and}(\mathrm{and}\ (cospl^{P\rightarrow t}\ (f_p\ x))(cap^{T\rightarrow t}\ (f_t\ x)))(\mathrm{port}^{L\rightarrow t}\ (f_l\ x)))$$

We now wish to apply this to the type T and to the transformations provided by the lexicon. No type clash with $cap^{T\rightarrow t}$, hence $id^{T\rightarrow T}$ works. For L and P we use the transformations f_p and f_l.

$$\begin{array}{l}(\mathrm{and}^{t\rightarrow(t\rightarrow t)}\\ \qquad(\mathrm{and}^{t\rightarrow(t\rightarrow t)}\\ \qquad\qquad(\mathrm{cospl}(f_p\ k^T)^P)^t)(\mathrm{cap}(\mathrm{id}\ k^T)^T)^t)^t(\mathrm{port}\ (f_l\ k^T)^L)^t)^t\end{array}$$



D Quantifiers: existential quantifier, indefinite and definite determiners



D.1. Usual Montagovian treatment

(3) There's a tramp sittin' on my doorstep (song)

(4) Some girls give me money (song)

(5) Something happened to me yesterday (song)

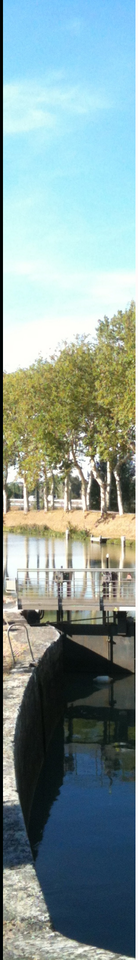
Usual view (e.g Montague)

Quantifier applies to the predicate,

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

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

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



D.2. Some problems of this usual treatment

(6) Keith played some Beatles songs.

(7) semantics: (some (Beatles songs)) (Keith played _)

(8) syntax (Keith (played (some (Beatles songs))))

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

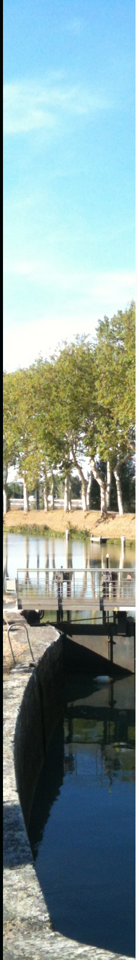
(9) Some politicians are crooks (web)

(10) ? Some crooks are politicians (us)

Same logical form.... how to interpret them differently.
This treatment of quantified NPs forces to dynamic binding (although E-type pronouns can be used)

(11) A man came in. He sat down.

A man by it self seems to be already denoting something.... (Russell, Geach,...)



D.3. Already a typed view?

At least, opposed to Frege's single sort view:

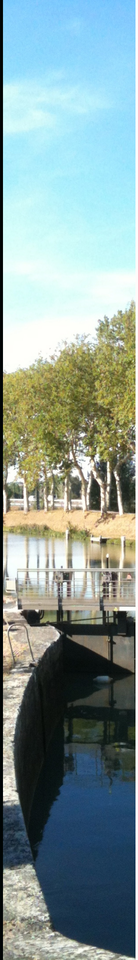
$$\exists{:}A\ P(x) \qquad \equiv \qquad \forall x.\ A(x) \& P(x)$$

$$\forall x{:}A\ P(x) \qquad \equiv \qquad \forall x.\ A(x) \rightarrow P(x)$$

(impossible for ”most of”)

in ancient and especially medieval philosophy (in particular Abu Barakat, Avicenna):

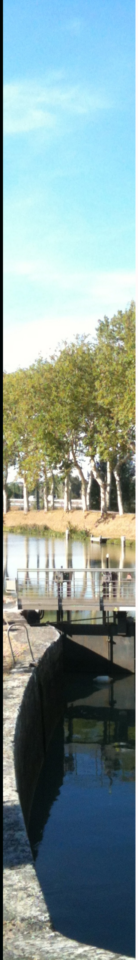
we assert properties of things as being member of some class (= type?)



D.4. A solution: Hilbert's epsilon

Given a formula F there is an individual term $\varepsilon_x F(x)$ (ε_x binds the occurrences of x in $F(x)$) with the intended meaning that $F(\varepsilon_x F) \equiv \exists x.\ F(x)$.

Follows syntactical structure. General presupposition $F(\varepsilon_x F)$: in natural language if we say "A man came in." we generally assert that such an individual exists and is a man.



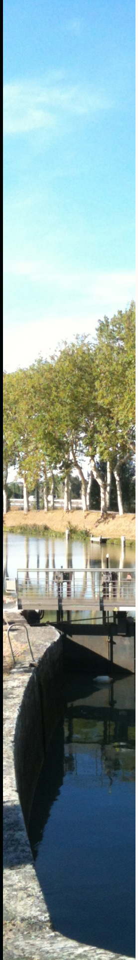
D.5. Definite and indefinite descriptions

$\iota_x F(x)$ The unique x such that (assumed to be due to Russell?)

According to von Heusinger (1995,1997, 2004) we can say the even if not unique, and it will pick up the most salient one, so *iota* can be left out.

- ε for definite descriptions
- η for indefinite description

Only a difference of interpretation: η a new one ε the most salient one.



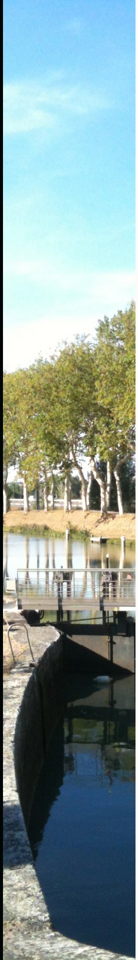
D.6. Other outcomes of Hilbert's operator

Universal quantifications: use τ ... with presupposition $P(\tau_x P(x))$

E-type pronouns:

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

(13) $''He'' = ''Aman'' = (\varepsilon_x M(x))$.



D.7. Typed Hilbert's operators? 1) base types.

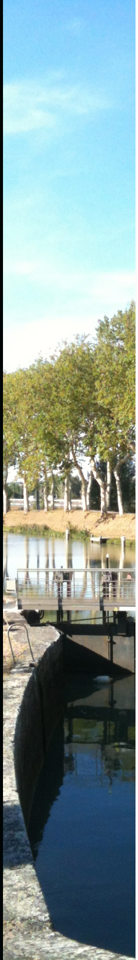
Personal intuition: there are less types than logical formulae with a single free variable, they are more constrained, they should be natural comparison classes.

What are the base types? CN (as Luo: rather sensible solution)

What about verbs and propositions?

(14) He did everything he could to stop them.

(15) And he believes whatever is politically correct and sounds good.

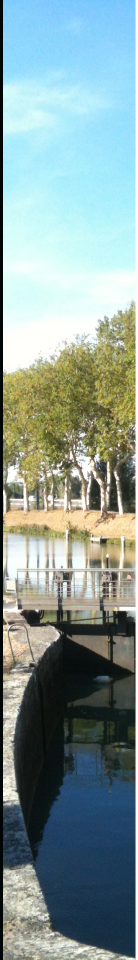


D.8. Typed Hilbert's operators? 2) predicates.

What is a predicate? what type is its domain?

For instance, is cat a property of animal or of all individuals ?

What relation between the type cat and the property of being a cat?



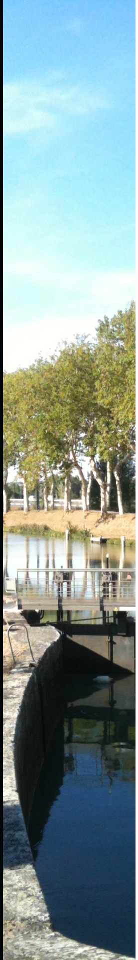
D.9. Some proposals

Predicates in a typed world:

- Simplest: predicates are of type $\mathbf{e} \rightarrow \mathbf{t}$ and they can be restricted.
- Predicates are not necessarily of type $\mathbf{e} \rightarrow \mathbf{t}$ (for being more natural) but a predicate $P^{\alpha\rightarrow\mathbf{t}}$ can be restricted and extended (they are false elsewhere). Hence, syntactically we do not need to know whether $\alpha \subset \beta$, $\alpha \subset \beta$

Types in a world with predefined predicates:

- Given a type α there is a corresponding predicate $\widehat{\alpha} : \mathbf{e} \rightarrow \mathbf{t}$. Too complicated to have a rule picking up the natural type "larger" than α. If types are some kind of an ontological tree, the mother node is good candidate.
- Alternative with, in my opinion, too many types: each formula with a single variable introduces a new type.



D.10. An example with an indefinite article

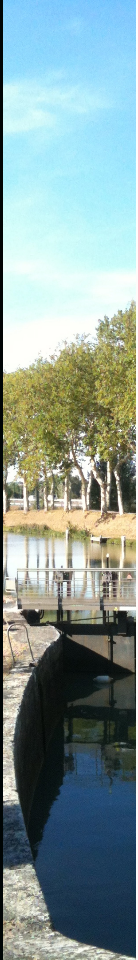
(16) a cat is sleeping under your car.

If cat is a property of type, say animal $\rightarrow \mathbf{t}$

- a: $\Lambda\alpha(\alpha \rightarrow \mathbf{t}) \rightarrow \alpha$ ("a" is a polymorphic ε). α gets instantiated as The general presupposition $P(a(P))$ yields $cat(a(cat))$. The epsilon term should be interpreted as an individual enjoying the property. If there is none, as an ideal individual, not in the denotation of P.

If cat is a type,

- It can be turned into a property and then do as above.
- a:$\Lambda\alpha.\alpha$ is strange but works (no problem to have a constant of type $\bot$) There is no need to add any presupposition, since because of the type of a one has $a(cat) : cat$.



D.11. Intepretation

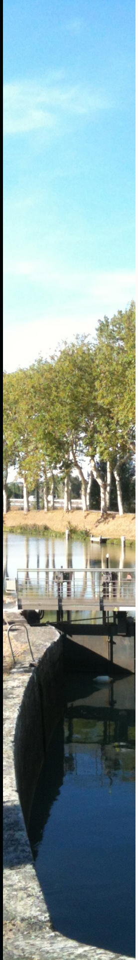
Interpretation of ε calculus is complicated Indeed, some ε-formulae have no equivalent formula in predicate logic. They can involve intricate dependencies, like Henkin branching quantifiers.

von Heusinger 's interpretation (IMHO: cosi cosi): a new one should be considered at each time. But...

(17) An old man came in he sat down. A tall man went out.

(18) $\varepsilon_x(M(x) \& O(x) \neq \varepsilon_x(M(x) \& T(x)$

Although they are not the same ε-terms, their referents ought to be different.



D.12. Definite description, universal quantifiers

The formalisation also works for definite descriptions:

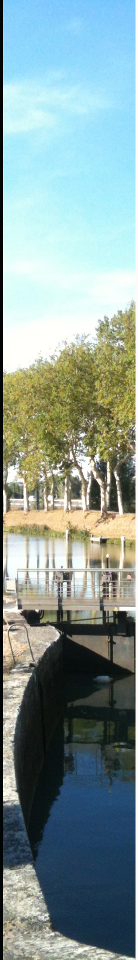
(19) The cat is sleeping under your car.

A difference is at interpretation: ι should be interpreted at the unique such that, and

In fact the simplest is the universal quantifier:

(20) Any cat sleeps a lot.

The presupposition $cat(\tau_x cat(x))$ (obtained from the general $P(\tau_x P(x))$)
The $\tau_x cat(x)$ is interpreted as an additional virtual element.



D.13. Generic NPs

How do we logically formulate ”most of” (much more than ”the majority of”) generic elements

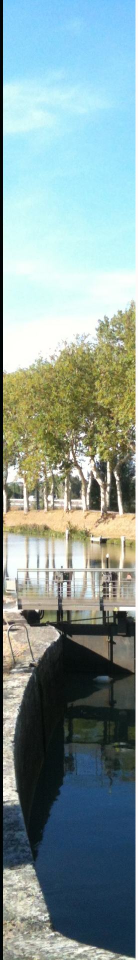
(21) The AKC notes that any dog may bite [...]

(22) The Brits love France.

(23) Un chien, ça peut toujours mordre.

idea to consider a fictive of fake element, like the τ and ε of Hilbert like the ι of van Heusinger.

(actually there is an other reading for 22 that we are just starting to think about: Brits love France more than similar classes do (Germans, Italians, etc.))

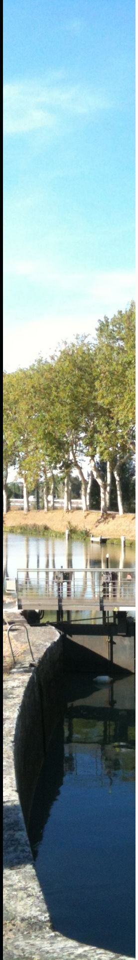


D.14. Radical minimalism / contextualism

Once we we appeal to comparison classes (a type) and its generic element we can address the following puzzle issued from Frege's view of a single domain:

- My daughter is tall and thin for a 2 year old.
- My two-year-old can't get his own cup [...] because he can't reach, [...]

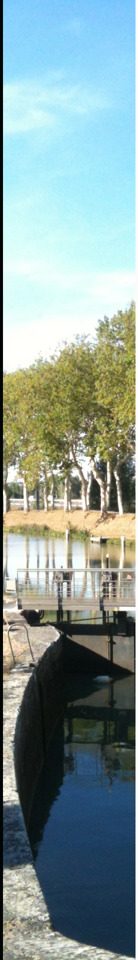
Carlotta who is a two year old girl it can be both tall and not tall, depending on her comparison class (her type in our type theoretic framework). Our type theoretical framework provides an account for such phenomena comparing Carlotta to the generic element of the corresponding class.



D.15. A personal view on the border between semantics and pragmatics

- semantics is encoded by the terms:
  they yield formulae by compositionality
- pragmatics is encoded in the types
  they are flexible and determined by the context

$$\forall\{\mathit{human}\}(\lambda x^{\mathit{human}}.\mathit{mortal}^{\mathit{human}\to\mathbf{t}}(x)).$$



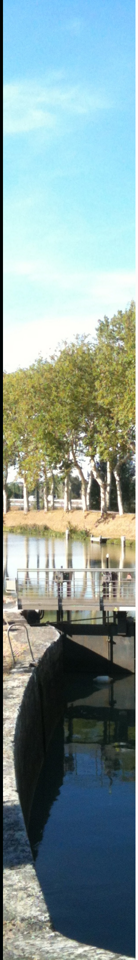
D.16. The syntax of ”most” generic elements

Observe that there is a difference between most (which refers to a cognitively accessible class) and ”mots of the ... that ...” which refers to a complex set, i.e. a formulae.

Consequently we shall have a first operator to obtain the generic from a type: **a constant** $\measuredangle$ **of type** $\Pi\alpha.\ \alpha$

When applied to a type T, this constant $\measuredangle$ yields the element $\measuredangle\{T\}$ of type T which is assumed to be the specimen of T: $\measuredangle$ maps each type to its specimen.

As opposed to standard work, we do not say that the generalised quantifier is a property of two predicates: indeed we are in a typed version, and the restriction predicate of the usual setting is the type.



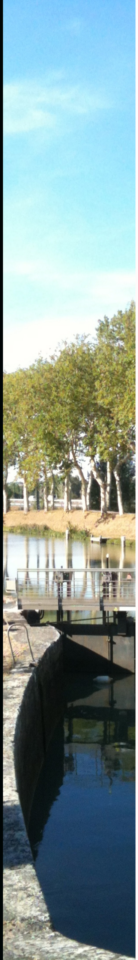
D.17. The operator for ”most of the ... that ...”

We need

another constant $\underline{\measuredangle}$ **of type** $\Pi\alpha.\ (\alpha \to \mathbf{t}) \to \alpha$

But it should be expressed at the same time that it is not an ordinary element of α but of the subset defined by the property of type $\alpha \to \mathbf{t}$.

In plain λ-calculus it is quite complicated, but in λ-DRT there are ways for such property to percolate on top level.



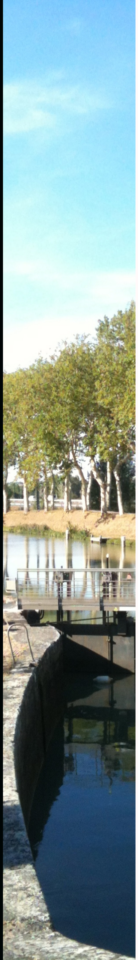
D.18. Being tall (as a child) and not tall (as a human being)

We have some term and functions, with standard types: **Carlotta** $\mathit{Carlotta} : 2\mathit{yoGirl}$ (constant) a class of child (these classes are vague)

$\mathsf{h} : 2\mathit{yoGirl} \rightarrow \mathit{human}$ (**optional λ-term**) these classes are included in the *human* class.

We can thereafter say that she is tall if she is taller than the average element in her class, an interval, and the class can be modified according to the context.

But the important point is that we can state such things and that they participate without any problem to the compositional process.



D.19. Being tall (as a child) and not tall (as a human being): computation

Here are the terms for:

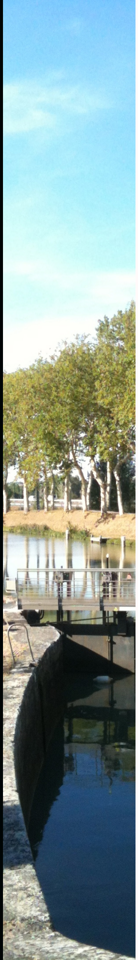
$height : \Pi\alpha.\ (\alpha \rightarrow float \rightarrow \mathbf{t})$

$<: float \rightarrow float \rightarrow \mathbf{t}$

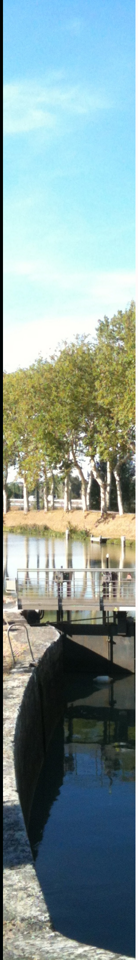
The term for *tall* below says that it is higher than any oh the heightS of the specimen. That's a possible view, to turn functions into relations for such an element.

tall $\Lambda\alpha.\lambda x^{\alpha}\forall\{float\}\lambda h_s^{float}\forall\{float\}\lambda h^{float}$
$height\{\alpha\}(\measuredangle\{\alpha\}, h_s) \wedge height\{\alpha\}(x, h) \Rightarrow h_s < h$

type of tall: $\Pi\alpha.\alpha \rightarrow \mathbf{t}$



E Other phenomena



E.1. Plurals

(24) John and Mary sneezed. (= John sneezed and Mary sneezed, OK).

(25) John and Mary met. ($\neq$ *John met and Mary met, OK).

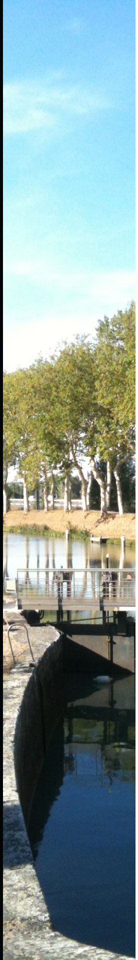
(26) The students wrote a paper. (“covering” reading: each student was part of group which wrote a paper)

(27) Each student wrote a paper. (no “covering” reading, OK)

(28) Three committees met. (two readings) OK

(29) (?) The committee sneezed (coercion?)

With operators (similar to the treatment of generalised quantification)



E.2. Virtual traveller (fictive motion)

(30) The path descended abruptly.

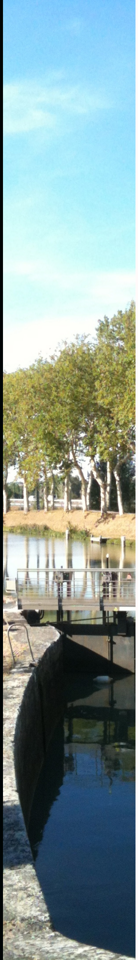
(31) The path descends for two hours.

(32) The tarred road runs along the coast for two hours.

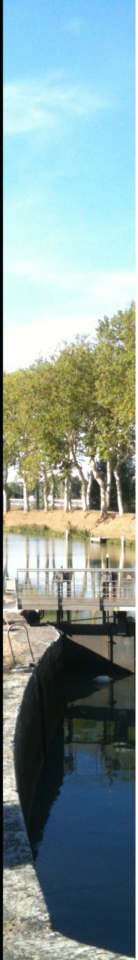
With transformation both from the verb and for the path...
be careful for the virtual traveller not to be tarred! There are trickier cases:

(33) The fence zigzags from the plateau to the valley.

(34) The highway crawls through the city.



F Conclusion



F.1. What we have seen so far

A **general framework** for

the logical syntax of **compositional semantics**

some lexical **semantics phenomena**

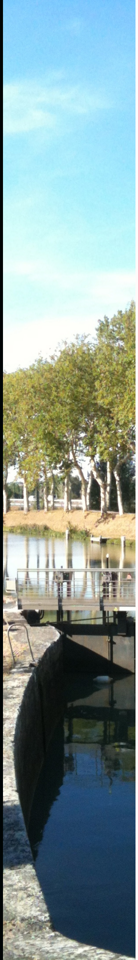
Guidelines:

Terms: semantics, sense, instructions for computing references

Types: pragmatics, defined from the context

Idiosyncratic (language specific) transformations compatible with the types but not forced by the types. J'ai crevé. / I went flat? Je suis venu à pieds, mon vélo est crevé / *déraillé.

Practically: some parts are implemented in Grail, Moot's wide coverage categorial parser, for fragment with a hand-typed semantic lexicon — but with λ-DRT instead of HOL in lambda calculus.



F.2. Perspective 1: base types, specialisation relations, subtyping

What are the base types?

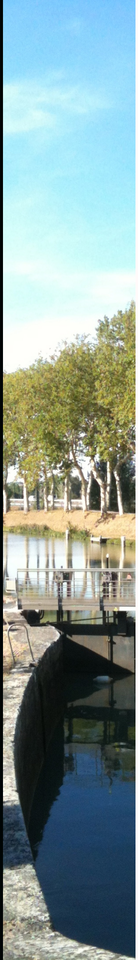
How can they be acquired?

Can the optional modifiers be acquired, at least the specialisation modifiers?

What about categorical interpretations (e.g. coherence spaces) as ontologies? (ontology-related question)

What would be an adequate notion of subtyping? (for a systematic coding of the ontological specialisation relations that are often admissible in the language).

Coercive sub typing of Luo & Soloviev can probably be extended to system F, work in progress with Sergeï. Coercive sub typing: **if** there is at most one coercion between any two base types **then** it remains so for any pair of types in the hierarchy with the induced coercions between complex types.



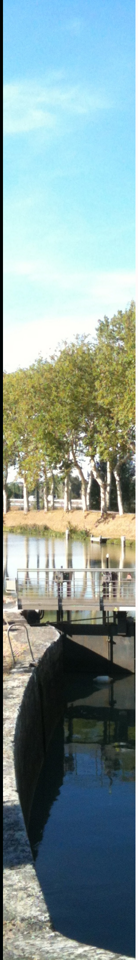
F.3. Perspective 2: formulae vs. types

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

Type to Formula: a type cat can be mirrored as a formula that can be refuted $\underline{\mathit{cat}} : \mathbf{e} \to \mathbf{t} \quad \underline{\mathit{cat}}(x) : \mathbf{t}$

Formula to Type? Is any formula with a single free variable a type? $\mathit{cat}(x) \wedge \mathit{belong}(x, \mathit{john}) \wedge \mathit{sleeps}(x)$:type?

At least it is not an implicit comparison class.



F.4. Some references

Work presented in this talk:

The Montagovian generative lexicon ΛTy_n: an integrated type-theoretical framework for compositional semantics and lexical pragmatics http://hal.archives-ouvertes.fr/hal-00779214

(with Michele Abrusci) Some proof theoretical remarks on quantification in ordinary language http://hal.archives-ouvertes.fr/hal-00779223

A lexicon for compositional semantics and lexical pragmatics: article in the Journal of Logic, Language and Information, 2010 (Ch. Bassac, B. Mery, Ch. Retoré)

Fictive motion, virtual traveller in French with plain Montagovian λ-terms (TALN 2011) or in English with λ-DRT (CID 2011) (R. Moot, L. Prévot, Ch. Retoré)

Quantification: ”most” in this setting (article in RLV) (Ch. Retoré)

Plurals: a talk at the Coconat workshop (R. Moot, Ch. Retoré)

Related work:

Nicholas Asher Lexical meaning in context: a web of words. Cambridge University Press 2011

Zhaohui Luo Contextual Analysis of Word Meanings in Type-Theoretical Semantics in Logical Aspects of Computational Linguistics 2012, Springer LNCS 6736 edited by S. Pogodalla and J-Ph. Prost.