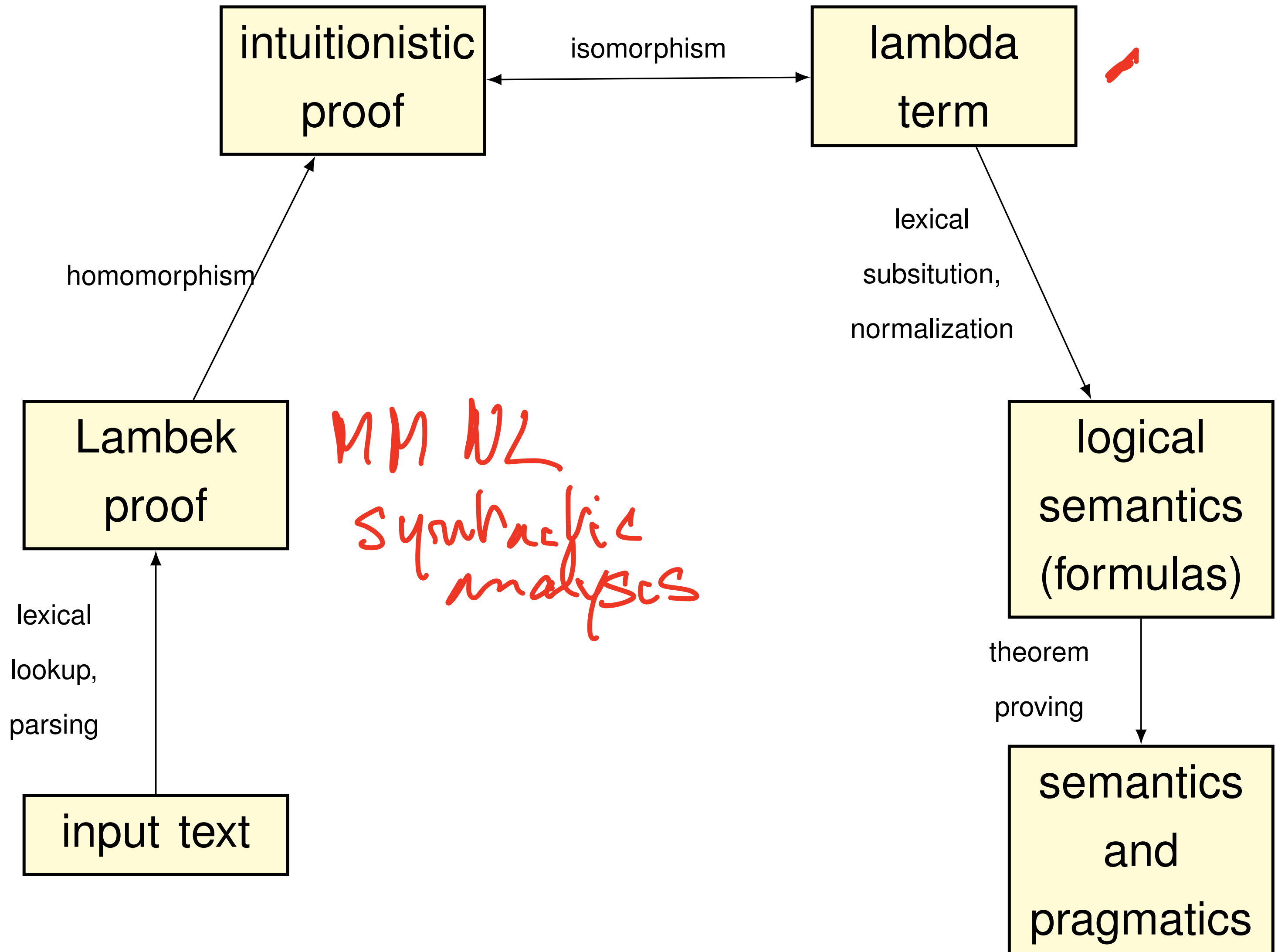
The Montagovian Generative Lexicon

Christian. Retoré

with Richard Moot
Bruno Mery
Livy Real

C.2. Architecture



	Introduction rules	Elimination rules
Intuitionistic	$\dfrac{\begin{matrix}[A]^n \\ \vdots \\ B\end{matrix}}{A \to B} \to I_n$	$\dfrac{A \quad A \to B}{B} \to E$
Lambek	$\dfrac{\begin{matrix}[A]^n \ldots \\ \vdots \\ B\end{matrix}}{A \backslash B} \backslash I_n$ $\dfrac{\begin{matrix}\ldots [A]^n \\ \vdots \\ B\end{matrix}}{B / A} / I_n$	$\dfrac{A \quad A \backslash B}{B} \backslash E$ $\dfrac{B / A \quad A}{B} / E$

C.3. Types and terms: Curry-Howard

A proof of $A \to B$ is a function that maps proofs of A to proofs of B.

Think of a formula/type as the set of its proofs.

Types are.... formulae.

λ-terms encode proofs $u : U$ means u is a term of type U.

We will also write $u : U$ as u^U.

C.4. Terms: Curry-Howard

1. *hypotheses* variables of each type which are terms of this type
2. *constants* there can be constants of each type
3. *abstraction* if $x : U$ is a **variable** and $t : T$ then $(\lambda x^U.\, t) : U \to V$.
4. *application* if $f : U \to V$ and $t : U$ then $(f\ t) : V$

With such typed terms we can faithfully encode proofs.

Variables are hypotheses (that are simultaneously cancelled).

C.5. Reduction and Normalisation

Reduction: $(\lambda x : U.\ t)^{U \to V} u^U$ reduces to $t[x := u] : V$.

Every simply typed lambda term reduces to a unique normal form, regardless the reduction strategy used.

C.6. Representing formulae within lambda calculus — connectives

Assume that the base types are $\mathbf{e}$ and $\mathbf{t}$ and that the only constants are

We need the following logical constants:

Constant	Type
$\exists$	$(e \to t) \to t$
$\forall$	$(e \to t) \to t$
$\wedge$	$t \to (t \to t)$
$\vee$	$t \to (t \to t)$
$\supset$	$t \to (t \to t)$

$(e \to t) \to t$ $e \to t$

Someone sleeps

t

C.7. Representing formulae within lambda calculus — language constants

The language constants for First Order Logic (for a start):

- R_q of type $\mathbf{e} \to (\mathbf{e} \to (.... \to \mathbf{e} \to \mathbf{t}))$
  e.g. likes: $e \to e \to t$, sleeps $e \to t$
- f_q of type $\mathbf{e} \to (\mathbf{e} \to (.... \to \mathbf{e} \to \mathbf{e}))$

C.8. Formulae and normal lambda terms

Proposition 4 *A normal lambda-term of type* $\mathbf{t}$ *using only the constants given above corresponds to a formula of first-order logic.*

C.9. Example: From formulae to normal lambda terms

$$\forall x.barber(x) \supset shaves(x,x)$$

$$\forall(\lambda x^{\mathbf{e}}.\ (\supset barber(x))((shaves(x))(x)))$$

Another one?

Detailed examples: a FOL formula as a term and as a natural deduction proof.

C.10. For Montague semantics

Non normal lambda terms of type $\mathbf{t}$ coming from syntax do not really correspond to formulae.

Hence we need:

- normalisation
- a proof that the normal terms do correspond to formulae, as we just shown.

C.11. Montague semantics. Types.

Simply typed lambda terms

$$types ::= e \mid t \mid types \rightarrow types$$

chair , *sleep* $e \rightarrow t$

likes transitive verb $e \rightarrow (e \rightarrow t)$

C.12. Montague semantics: Syntax/semantics.

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

Logical operations (and, or, some, all the,.....) are the lambda-term constants defined above.

C.13. Montague semantics
Logic within lambda-calculus

Words in the lexicon need constants for their denotation:

likes	$\lambda x \lambda y\ (\mathrm{likes}\ y)\ x$	$x : e,\ y : e, \mathrm{likes} : e \rightarrow (e \rightarrow t)$
<< likes >> is a two-place predicate		
Garance	$\lambda P\ (P\ \mathrm{Garance})$	$P : e \rightarrow t,\ \mathrm{Garance} : e$
<< Garance >> is viewed as the properties that << Garance >> holds		

C.14. Montague semantics. Computing the semantics 1/5

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

word	syntactic type u semantic type u^* semantics: λ-term of type u^* x^v means that the variable or constant x is o
some	$(s/(np\backslash s))/n$ $(e \to t) \to ((e \to t) \to t)$ $\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}(P\ x)(Q\ x$
statements	n $e \to t$ $\lambda x^{e}(\mathtt{statement}^{e\to t}\ x)$
speak_about	$(np\backslash s)/np$ $e \to (e \to t)$ $\lambda y^{e}\ \lambda x^{e}\ ((\mathtt{speak\_about}^{e\to(e\to t)}\ x)\ y)$
themselves	$((np\backslash s)/np)\backslash(np\backslash s)$ $(e \to (e \to t)) \to (e \to t)$ $\lambda P^{e\to(e\to t)}\ \lambda x^{e}\ ((P\ x)\ x)$

C.15. Syntactic proof

Let us first show that "*Some statements speak about themselves*" belongs to the language generated by this lexicon. So let us prove (in natural deduction) the following:

$$(s/(np\backslash s))/n\ ,\ n\ ,\ (np\backslash s)/np\ ,\ ((np\backslash s)/np)\backslash(np\backslash s) \vdash s$$

some statement speaks about themselves

$$\dfrac{\dfrac{(s/(np\backslash s))/n \quad n}{(s/(np\backslash s))}\,/E \qquad \dfrac{(np\backslash s)/np \quad ((np\backslash s)/np)\backslash(np\backslash s)}{(np\backslash s)}\,\backslash E}{s}\,/E$$

C.16. Syntactic Proof to Semantic proof

$$\dfrac{\dfrac{(s/(np\backslash s))/n \quad n}{(s/(np\backslash s))}/E \quad \dfrac{(np\backslash s)/np \quad ((np\backslash s)/np)\backslash(np\backslash s)}{(np\backslash s)}\backslash E}{s}/E$$

Using the homomorphism from syntactic types to semantic types we obtain the following intuitionistic deduction.

$$\dfrac{\dfrac{(e\rightarrow t)\rightarrow(e\rightarrow t)\rightarrow t \quad e\rightarrow t}{(e\rightarrow t)\rightarrow t}\rightarrow E \quad \dfrac{e\rightarrow e\rightarrow t \quad (e\rightarrow e\rightarrow t)\rightarrow e\rightarrow t}{e\rightarrow t}\rightarrow E}{t}\rightarrow E$$

C.17. Semantic Proof to Lambda Term

$$\dfrac{\dfrac{(e \to t) \to (e \to t) \to t \quad e \to t}{(e \to t) \to t}\to E \qquad \dfrac{e \to e \to t \quad (e \to e \to t) \to e \to t}{e \to t}\to E}{t}\to E$$

$$\dfrac{\dfrac{So^{(e\to t)\to(e\to t)\to t} \quad Sta^{e\to t}}{(So\ Sta)^{(e\to t)\to t}}\to E \qquad \dfrac{SpA^{e\to e\to t} \quad Refl^{(e\to e\to t)\to e\to t}}{(Refl\ SpA)^{e\to t}}\to E}{((So\ Sta)\ (Refl\ SpA))^{t}}\to E$$

C.18. Montague semantics. Computing the semantics. 3/5

The syntax (e.g. a Lambek categorial grammar) yields a λ-term representing this deduction simply is

((some statements) (themselves speak_about)) of type t

C.19. Montague semantics. Computing the semantics. 4/5

$$\Big((\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^e(\wedge(P\ x)(Q\ x)))))$$ same

$$(\lambda x^e(\mathtt{statement}^{e\to t}\ x))\Big)$$

$$\Big((\lambda P^{e\to(e\to t)}\ \lambda x^e\ ((P\ x)x))$$ themselves

$$(\lambda y^e\ \lambda x^e\ ((\mathtt{speak\_about}^{e\to(e\to t)}\ x)y))\Big)$$

$$\downarrow\beta$$

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

$$(\lambda x^e\ ((\mathtt{speak\_about}^{e\to(e\to t)}\ x)x))$$

$$\downarrow\beta$$

$$(\exists^{(e\to t)\to t}\ (\lambda x^e(\wedge(\mathtt{statement}^{e\to t}\ x)((\mathtt{speak\_about}^{e\to(e\to t)}\ x)x))))$$

C.20. Montague semantics. Computing the semantics. 5/5

This term represent the following formula of predicate calculus (in a more pleasant format):

$$\exists x : e\ (\mathtt{statement}(x) \land \mathtt{speak\_about}(x, x))$$

This is a (simplistic) semantic representation of the analysed sentence.

What about
Lexical Semantics?

How can we deal with
polysemy
many senses
facets of one sens?

D.1. Examples of Lexical Issues

Short roadmap:

- Restriction of selection, polysemy, felicity
- System-F and our framework
- Determiners and quantification
- Classical GL constructions
- Co-predication and constraints
- Deverbals
- Fictive motion
- Integrating plurals and their readings
- Specific issues

D.2. Restriction of Selection and Polysemy

Selection

- Predicates (syntactically) select arguments
- The lexical field of those arguments is restricted
- Other arguments can be forced to behave as expected

Differences in acceptability

- The dog barked.
- The chair barked.
- The drill sergeant barked.
- The hawker barked.

D.3. Acceptability and Felicity: Semantics, Pragmatics or Both ?

Montague: everything is acceptable

- All syntactically valid items have the same semantic “meaning”
- We have to rely on pragmatics or interpretation

Too strong restriction from lexical semantics

- **e** is replaced by many sorts
- Barking dogs are licensed, everything else is blocked
- No language works that way

Creative uses and semantic licenses

- Fast runners, cars, computers... and phones
- A delicious game (Cooper)
- Expertly built (Adams)

intersection types
$\cup$
F

D.4. System F

Types:

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

Terms

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

The reduction is defined as follows:

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

D.8. Basic facts on system F

Logicians / philosophers often ask whether system F is safe?

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

much better than D, intersection types

- Confluence and strong normalisation — requires the comprehension axiom for all formulae of HA_2. (Girard 1971)
- A concrete categorical interpretation with coherence spaces that shows that there are distinct functions from A to B.
- Terms of type **t** with constants of mutisorted FOL (resp. HOL) correspond to multisorted formulae of FOL (resp. HOL)
- Possiblilty to have **coercive sub typing** for ontological inclusion (cats are animals etc.)

D.9. Examples of second order usefulness

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

Polymorphic conjunction:

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

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

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

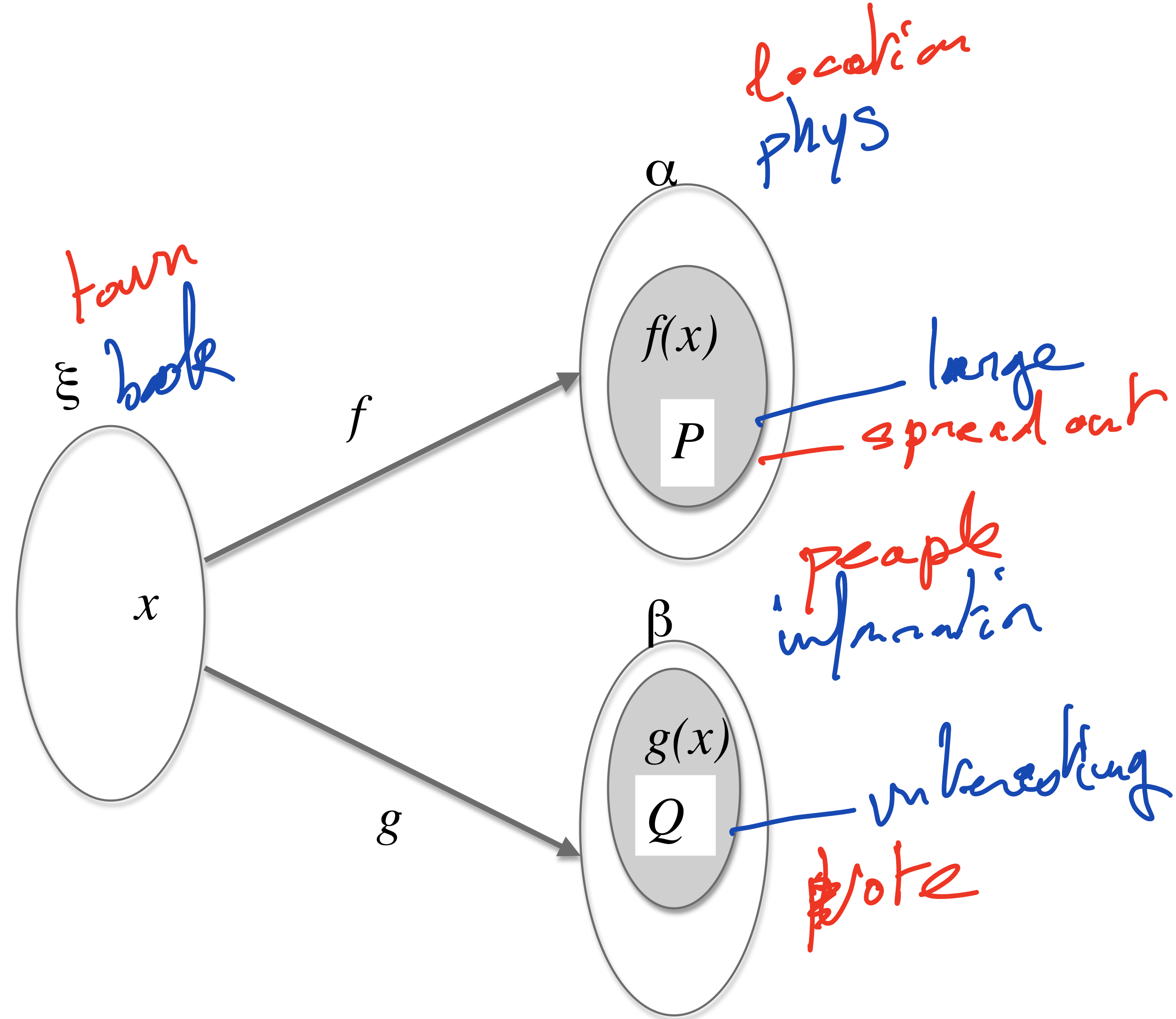


Figure 1: Polymorphic conjunction: $P(f(x))\&Q(g(x))$
with $x:\xi$, $f:\xi\rightarrow\alpha$, $g:\xi\rightarrow\beta$.

MTT

D.10. Coercive subtyping for F (Luo, Soloviev, Retoré)

F

Key property: at most one coercion between any two types.

Given coercions between base types.

Propagates through type hierarchy (unique possible restoration).

coercive application $$\frac{f : A \to B \quad u : A_0 \quad A_0 < A}{(f\ a) : B}$$

....

$$\frac{A < B \quad C < D}{B \to A < C \to D} \qquad \frac{A < B}{X \to A < X \to B} \qquad \frac{A < B}{B \to X < A \to X}$$

....

$$\frac{U < T[X]}{U < \Pi X.\ T[X]}\ X \text{ not free in } U \qquad \frac{U < \Pi X.\ T[X]}{U < T[W]}$$

D.11. Coercive subtyping

Theorem: [hierarchical coherence] Whenever the above system derives $a < b$ where a and b are base types the coercion $a < b$ was a given coercion.

Coercive sub typing seems adequate to model ontological inclusions in particular between base types A car is a vehicle

These morphisms / coecions are identity on the object, hence only one morphism may exist between two given base types.

Possibly there are more coercions than ontological inclusions.

D.12. In a Nutshell

The Generative Lexicon

- Pustejovsky, 1995 (and precursors)
- Discussed and refined by Asher, Cooper, Luo...
- Idea: the lexicon provides enough data to generate word meanings in context

optional λ terms

Framework sketch

- Types (and terms) from System-F
- Lexical entries are typed with many sorts
- Each word has a single main λ-term
- Each word can have any number of optional λ-terms
- Those terms are transformations, and are word-based
- Normal application is the same
- Transformations are used when types clash
- Types guide the selection of transformations

$W \neq U$

optional terms
linked to word
not to types

D.13. Lexicon v. Type

Why do we think transformations are lexical ?

(Rather than type-driven)

In a word: idiosyncrasy.

Consider:

(5) La classe est finie. (Event)
(6) La classe est fermée. (Location)
(7) La classe est de bon niveau. (People)
(8) La promotion est de bon niveau. (univoque: People)

In French, the two words do not have the same possible uses, but represent exactly the same group of people.

(This also seems to be the case in American English.)

D.14. Strong Idiosyncrasy

Linguistic constructs are not independent of the language

(Pleonasm ?)
Idioms and specific constructs are illustrations of this.

Differences of language

I have punctured

Differences of dialect

Un demi-fraise

Differences of jargon

Redresse la #16

D.15. Toy Example

Named towns are examples of highly polysemous words that can be referred to for their location, population, and many other aspects.

- Types : T (town), Pl (place), P (people)
- Usual predications:
  1. Birmingham is spread out
  2. Birmingham voted labour
  3. 1 & 2

Lexical item	Main λ-term	Modifiers
Birmingham	$birmingham^T$	$Id_T : T \to T$ $t_2 : T \to P$ $t_3 : T \to Pl$
is_spread_out	$spread\_out : Pl \to \mathbf{t}$	
voted	$voted : P \to \mathbf{t}$	

people
place

1. Type mismatch in $spread\_out^{Pl\rightarrow \mathbf{t}}(Birmingham^{T})$, resolved using t_3:

$$spread\_out^{Pl\rightarrow \mathbf{t}}(t_3{}^{T\rightarrow Pl} Birmingham^{T}))$$

2. The same, using t_2:

$$voted\, P \rightarrow \mathbf{t}(t_2{}^{T\rightarrow Pl} Birmingham^{T})$$

3. We use a polymorphic conjunction operator, $\&^{\Pi}$.

   –as seen.

$$\Lambda\xi\lambda x^{\xi}\, \lambda f^{\xi\rightarrow\alpha} \lambda g^{\xi\rightarrow\beta}(\mathrm{and}^{(\mathbf{t}\rightarrow\mathbf{t})\rightarrow\mathbf{t}}\, (spread\_out\ (f\ x))(voted\ (g\ x)))$$

After application, we have:

$$(\mathrm{and}\ (spread\_out^{Pl\rightarrow\mathbf{t}}\ (t_3{}^{T\rightarrow Pl}\ Birmingham^{T}))$$

$$(voted^{Pl\rightarrow\mathbf{t}}\ (t_2{}^{T\rightarrow P}\ Birmingham^{T})))$$

$\rightsquigarrow$ Sp(B) & voted(B)

D.35. Dot Objects

”Dots” are very special objects in GL. Examples:

- The book was heavy and interesting.
- The lunch was delicious but took forever.
- The pressure is 120 psi and rising.
- The fair city of Perth, by the river Tay, is a bustling shopping and trade centre that nevertheless retains a tranquil atmosphere. . .

We do not differentiate between qualia, dot ”facets” and provide transformations for everything, such as

$$f_{Phys}{}^{Book \to \varphi}, f_{Info}{}^{Book \to I}$$

D.36. Constraints

Possible/Hazardous co-predicative constructions

- Important point: this is not (only) about toy examples.
- *The salmon was fast and delicious.
- The salmon was lightning fast. It is delicious.
- Birmingham is a large city and voted labour.
- *Birmingham is a large city and won the cup.

Constraints on flexibility: FLEXIBLE v. RIGID

- Are lexically fixed on modifiers, computed for terms
- FLEXIBLE: anything goes.
- RIGID: nothing else can go (even the original typing).

Town → Football Club: is rigid

Complex example

Lexicon

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

where the base types are defined as follows:

T	town
F	football club
P	people
Pl	place

Birmingham is spread out and won

Polymorphic AND yields:

$(\&^{\Pi}(spread\_out)^{Pl \to \mathbf{t}}(won)^{P \to \mathbf{t}})$

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

$$\&^{\Pi}\{Pl\}\{P\}(spread\_out)^{Pl \to \mathbf{t}}(won)^{P \to \mathbf{t}}$$

T —> Fooh

It reduces to:

$\Lambda\xi\lambda x^{\xi}\ \lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}(\mathrm{and}^{\mathbf{t}\to\mathbf{t})\to\mathbf{t}}\ (spread\_out\ (f\ \ x))(won\ (g\ x)))$

Should apply to t_1 and t_3 – but t_1 is RIGID.

Syntactical relaxation of semi-flexible constraints

- The salmon was lightning fast. It is delicious.
- Semi-flexible: acts as Rigid, but is reset by reference.

Using this setting we modelled:

- counting puzzle

  I carried the books to the attic, because I read them all

- deverbals (with Livy Real)

  the signature cannot be read

  the signature took ages

- fictive motion

  "the path descends for half an hour

But also formal semantic issues

- quantification in a many sorted environment with Hilbert ε, τ,
- plurals

The committee met,
The team lost.
The team had covid.

System F $\mathbb{N}$ count.

AMGL acquisition ???

machine learning of λ terms??

extracting information
from a large lexical semantic
network

JeuxDeMots

4 millions of terms
100 millions of
labelled relations

Thanks to Richard Moot
implementation of Grail
categorial parser

MMCG $\longrightarrow$ λ DRT

(supertagging
proofnet / parse structure) DEEP LEARNING
(efficiency $\uparrow$)

D.49. The Differences between our Proposal and Related Formulations

Ontological Types

- Most other approaches
- Asher, Bekki, Pustejovski. . .
- Ontology (concept) provides types
- Types provide all adaptations (co-compositions, shifts, accommodations. . .)

Lexical Sorts ∼ classifiers (Sign languages
African languages
Chinese, Japanese

- Basis for our approach
- Types provide a mechanism for recognising clashes
- Transformations come from the lexicon
- Idiosyncrasies in languages and dialects are possible
- Closer to the linguistic data

Meaning computation is POLYNOMIAL (β reduction in soft linear logic)

Our approach

- Based on the latter
- General type-based transformations are still used
- The lexicon can overload type transformations
- (Simply for reasons of scale and practicality)

Other possibilities

- Nunberg & Sag, transfers of meaning
- Cooper: type records
- Luo: words as types

PAPERS ON THE MONTAGOVIAN GENERATIVE LEXICON
Most of them areavaailable from https://www.lirmm.fr/~retore/
In bold the ones I consider more important or relevant to the workshop

- **2007** Christian Bassac Bruno Mery Christian Retoré A Montagovian generative lexicon in Formal Grammar 2007.
- 2010 Christian Bassac, Bruno Mery, Christian Retoré Towards a Type-Theoretical Account of Lexical Semantics Journal of Logic, Language and Information. 19(2) pp. 229—245 2010 https://doi.org/10.1007/s10849-009-9113-x
- **2014 Christian Retoré The Montagovian Generative Lexicon ΛTy_n: a Type Theoretical Framework for Natural Language Semantics in Ralph Matthes and Aleksy Schubert (eds) 19th International Conference on Types for Proofs and Programs (TYPES 2013) LIPICS vol 26 pp. 202--229. http://dx.doi.org/10.4230/LIPIcs.TYPES.2013.202**
- **2014 Livy Real, Christian Retoré Deverbal semantics and the Montagovian generative lexicon *Journal of logic, language and information Vol. 23 n. 3 pp. 347-366* http://dx.doi.org/10.1007/s10849-014-9187-y**
- **2015** Bruno Mery and Christian Retoré Are Books Events? Ontological Inclusions as Coercive Sub-Typing, Lexical Transfers as Entailment in Eric McReady (ed) LENLS12, Tokyo, November 2015. pp. 74-87
- **2015** Livy Real and Christian Retoré A Case Study of Copredication over a Deverbal that Reconciles Empirical Data with Computational Semantics in Eric McReady (ed) LENLS12, Tokyo, November 2015. pp. 54-66.
- **2015 Bruno Mery, Richard Moot, Christian Retoré Computing the Semantics of Plurals and Massive Entities using Many-Sorted Types In Murata, Tsuyoshi and Mineshima, Koji and Bekki, Daisuke (Eds) New Frontiers in Artificial Intelligence LNCS 9067 Springer 2015. pp. 144--159 http://dx.doi.org/10.1007/978-3-662-48119-6_11**
- 2017 Bruno Mery and Christian Retoré Classifiers, Sorts and Base types in the Montagovian Generative Lexicon and other Type Theoretical Frameworks for Lexical Semantics in Stergios Chatzikyriakidis and Zhaohui Luo (eds) *Modern Perspectives in Type-Theoretical Semantics,* Springer.2017. http://dx.doi.org/10.1007/978-3-319-50422-3
- **2018 Mathieu Lafourcade, Bruno Mery, Mehdi Mirzapour, Richard Moot and Christian Retoré *Collecting Crowd-Sourced Lexical Coercions for Compositional Semantic Analysis* in Arai, S., Kojima, K., Mineshima, K., Bekki, D., Satoh, K., Ohta, Y. (Eds.) *New Frontiers in Artificial Intelligence* Springer LNCS 10838 pp. 214-230 2018.**
- 2019 Anaïs Lefeuvre-Halftermeyer, Richard Moot, and Christian Retoré *A computational account of virtual travelers in the Montagovian generative lexicon.* In D. Stosic and M. Aurnague (eds) The Semantics of Dynamic Space in French. John Benjamins pp. 324-352. 2019 https://doi.org/10.1075/hcp.66.09lef
- ***2019 Bruno Mery, Richard Moot, Christian Retoré Solving the Individuation and Counting Puzzle with λ-DRT and MGL In: Kojima K., Sakamoto M., Mineshima K., Satoh K. (eds) New Frontiers in Artificial Intelligence. JSAI-isAI 2018. Lecture Notes in Computer Science, vol 11717. Springer, pp. 298-312 https://doi.org/10.1007/978-3-030-31605-1_22***
- **2019 Richard Moot and Christian Retoré Natural Language Semantics and Computability Journal of Logic, Language and InformationVolume 28, Issue 2, pp 287–307 https://doi.org/10.1007/s10849-019-09290-7**