Sémantique et modélisation dans le cadre de la logique catégorique

Christian Retoré
(Université de Bordeaux, LABRI-CNRS & INRIA)

PLAN

Faute de temps de préparation,

deux petits exposés plutôt qu'un grand uni:

- Un petit panorama de méthodes logiques pertinentes mais plus ou moins bien connues en sémantique formelle et notamment:
  - **La théorie des types intuitionniste et son interprétation catégorique**
- Une utilisation spécifique pour une sémantique plus fine des briques de bases de la sémantique compositionnelle, à savoir les mots.
  - **Un lexique sémantique à la Montague mais qui rende compte des facettes d'un sens, de l'adaptation du sens, etc**

DÉMONSTRATIONS, MODÈLES, COMPLÉTUDE: LE CAS CLASSIQUE

- Logique classique
  - Propositionnelle (pvq) & r v ¬ p
    - Valeurs de vérités: intepprétation, tables de vérité
    - Preuves Hilbert / Séquents
    - Démontrable ⇔ vraie pour toute interprétation
  - Premier ordre ⋀x ⋁(Px v ¬Pf(x,y))
    - Ensemble E
    - Prédicat à n places : partie de En
    - Fonction à n arguments : fonction de En dans E
    - Preuves Hilbert / séquents
    - Démontrable ⇔ vraie dans tout modèle pour toute interprétation.

Démonstrations classiques et intuitionnistes

Règles structurelles

$$\frac{\Gamma, A, B, \Delta \vdash \Theta}{\Gamma, B, A, \Delta \vdash \Theta}\,E_g \qquad \frac{\Theta \vdash \Gamma, A, B, \Delta}{\Theta \vdash \Gamma, B, A, \Delta}\,E_d$$

$$\frac{\Delta \vdash \Theta}{A, \Delta \vdash \Theta}\,E_g \qquad \frac{\Theta \vdash \Gamma}{\Theta \vdash \Gamma, A}\,E_d$$

$$\frac{\Gamma, A, A, \Delta \vdash \Theta}{\Gamma, A, \Delta \vdash \Theta}\,C_g \qquad \frac{\Theta \vdash \Gamma, A, A, \Delta}{\Theta \vdash \Gamma, A, \Delta}\,C_d$$

DÉDUCTION NATURELLE OU LAMBDA CALCUL TYPÉ

Règles logiques

$\dfrac{\Theta \vdash (A \wedge B)}{\Theta \vdash A}\wedge_e \quad \dfrac{\Theta \vdash (A \wedge B)}{\Theta \vdash B}\wedge_e$	$\dfrac{\Theta \vdash A \quad \Delta \vdash B}{\Theta, \Delta \vdash (A \wedge B)}\wedge_i$
$\dfrac{\Theta \vdash (A \vee B) \quad A, \Gamma \vdash C \quad B, \Delta \vdash C}{\Theta, \Gamma, \Delta \vdash C}\vee_e$	$\dfrac{\Theta \vdash A}{\Theta \vdash (A \vee B)}\vee_i \quad \dfrac{\Theta \vdash B}{\Theta \vdash (A \vee B)}\vee_i$
$\dfrac{\Theta \vdash A \quad \Gamma \vdash A \Rightarrow B}{\Gamma, \Theta \vdash B}\Rightarrow_e$	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash (A \Rightarrow B)}\Rightarrow_i$
$\dfrac{\Gamma \vdash \perp}{\Gamma \vdash C}\perp_e$	

SÉQUENTS

Règles

$$\frac{\Gamma, A, B, \Delta \vdash \Theta}{\Gamma, (A \wedge B), \Delta \vdash \Theta}\,\wedge_g \qquad \frac{\Theta \vdash \Gamma, A, \Delta \qquad \Theta \vdash \Gamma, B, \Delta}{\Theta \vdash \Gamma, (A \wedge B), \Delta}\,\wedge_d$$

$$\frac{\Gamma, A, \Delta \vdash \Theta \qquad \Gamma, B, \Delta \vdash \Theta}{\Gamma, (A \vee B), \Delta \vdash \Theta}\,\vee_g \qquad \frac{\Theta \vdash \Gamma, A, B, \Delta}{\Theta \vdash \Gamma, (A \vee B), \Delta}\,\vee_d$$

$$\frac{\Gamma, \Delta \vdash A, \Theta \qquad B, \Gamma, \Delta \vdash \Theta}{\Gamma, (A \Rightarrow B), \Delta \vdash \Theta}\,\Rightarrow_g \qquad \frac{A, \Gamma \vdash B, \Delta, \Theta}{\Gamma \vdash \Delta, (A \Rightarrow B), \Theta}\,\Rightarrow_d$$

$$\frac{\Gamma, \Delta \vdash A, \Theta}{\Gamma, (\neg A), \Delta \vdash \Theta}\,\neg_g \qquad \frac{A, \Gamma \vdash \Delta, \Theta}{\Gamma \vdash \Delta, (\neg A), \Theta}\,\neg_d$$

On peut ajouter la règle dite de coupure (qui correspond grosso modo à l'utilisation d'un lemme) :

$$\frac{\Theta \vdash \Gamma, A \qquad A, \Phi \vdash \Psi}{\Theta, \Phi, \Theta\Gamma, \Psi}\,\wedge_d$$

QUANTIFICATION

Règles pour les quantificateurs

$\dfrac{\Gamma \vdash (\forall x \quad A(x))}{\Gamma \vdash A(t)}\,\forall_e$	$\dfrac{\Theta \vdash A(y)}{\Theta \vdash (\forall x \quad A)}\,\forall_d$ plus de y libre ensuite
$\dfrac{\Gamma \vdash (\exists x \quad A(x)) \quad A(y), \Delta \vdash C}{\Gamma, \Delta \vdash C}\,\exists_g$ plus de y libre ensuite	$\dfrac{\Theta \vdash A(t/x)}{\Theta \vdash (\exists x \quad A(x))}\,\exists_d$

RECTIFICATION

- La déduction naturelle est « naturellement » intuitionniste. Sinon il faut ajouter Av¬A
- Preuve intuitionniste = lambda termes
- Lambda terme de type t = formule (cf. Church, Montague)
- ??????????
- Lambda terme = preuve propositionnelle de la bonne formation de la formule, permettant des substitutions.

Modèle pour la Logique intuitionniste du premier ordre: la catégorie des Faisceaux de modèles (vérité locale)

- Espace topologique X, sous ens. Ouverts
  - Clos par union quelconques
  - Clos par intersections finies
- Un modèle usuel M_U pour chaque ouvert U
- Restriction préservant l'interprétation
- Condition de recollement si coïncidence sur les intersection, il existe un élément global (préfaisceau de modèles -> faisceau de modèles)
- Vérité dans un ouvert définie par *forcing* par ex: s
  - AvB vrai sur U s'il existe U_A U_B, $U_A \cup U_B = U$ A vrai sur U_A et B vrai sur U_B
  - $\vee$x A(x) il existe un recouvrement U_i et dans chaque M_{ui} il existe d tel que A(d)

Modèle pour la Logique intuitionniste du premier ordre
Faisceaux de modèles $= \Omega$ ensembles

- Ω = algèbre de Heyting complète (sorte d'algèbre de Boole, formules quotientées par la prouvabilité) $A\text{->}B = \bigvee \{C \mid C \,\&\, A \leq B\}$
- M un ensemble, des éléments partiels E(x)= [[x=x]] , valeur de vérité dans Ω : $P(a) \leq E(a)$ etc.
- Comme d'hab. la complétude vient en prenant Ω = formules quotientées par l'équivalence
- Remarque: tout espace topologique donne une algèbre de Heyting complète (réciproquement les algébres de Heyting complètes sont plus générale que les espaces topologiques, topologies de Grothendieck)

MODÈLES DE LA LOGIQUE INTUITIONNISTE, VOIRE MODALE

- Complétude: démontrable en intuitionnisme ⇔ vrai dans tout faisceau de modèle (même en fixant l'espace topologique à 2^{ω})
- Faisceaux préférables aux modèles de Kripke:
  - Il en existe dans la « nature » mathématique
  - Plus contraints -> beaucoup de propriétés.
- Fonctionnent pour LJ et donc pour S4 (intertraductibles)
- On peut faire des constructions de faisceaux directes, pour mieux gérer les références variables: *A First Order Modal Logic and its Sheaf Models* (Barnaby P. Hilken and David E. Rydeheard)
- Utilisés par Petitot, pour rendre compte de propriétés dans l'espace qui n'ont pas de bord.

MODÈLES USUELS POUR LE SECOND ORDRE ET ORDRE SUPÉRIEUR

- On souhaite parfois utiliser des variables de prédicats, de prédicats de prédicats, etc. et quantifier sur elles.
- Ça change quoi? Au lieu de faire varier dans toutes les parties (parties de parties, etc.) du modèle, il faut faire varier dans toute sous algèbre de Boole/Heyting de parties du modèles. (celles qui sont définissables)
- Une partie d'un ensemble, c'est une fonction de l'ensemble dans Ω (ceux qui s'envoient sur 1).

CHANGEMENT DE PERSPECTIVE: MODÈLES DES PREUVES, CATÉGORIES CARTÉSIENNES FERMÉES

- Catégorie
  - objets A, B, …~ensembles, structures, A, B, etc.
  - morphismes de A dans B: <A;B>
    (en général application préservant la structure)
- Logique, preuves
  - Objets = formules
  - Morphismes = preuves lambda termes / beta
    Si tous les morphismes sont équivalents
    -> ordre partiel algèbre de Boole ou de Heyting.

- & produit
- => objet espace des fonctions
- 0 initial
- 1 final

Pas de sémantique catégorique pour la logique classique (Joyal)

- ¬X = (X=>0)
- <(0&A);B > ≅ <0;(A=>B)> 1 élt. 0 initial
- Donc (0&A)≅ 0 et en particulier 0&0 ≅ 0 π1=π2
- Soit h,k: <C,0> on a
  - h=π1<h;k>:<C&C;0&0>
  - k=π2<h;k>:<C&C;0&0>
  - π1=π2
  - h=k
  - Au plus un morphisme vers 0
- <A;B>≅<B=>0;A=>0>≅<(B=>0)&A;0>
- Au plus un morphisme entre deux objets
- Au plus une preuve de toute formule!

MODÈLE CATÉGORIQUE PROPOSITIONNEL EXEMPLE CONCRET: ESPACES COHÉRENTS

- Graphes (infinis) objets: cliques
- Fonctions stables
  - Commutent aux unions dirigées
  - Aux intersections finies dans une clique
- On peut faire des produits (conjunction)
- L'espace des fonctions stables de A dans B est représentable comme un espace cohérent: c'est l'implication
- Les preuves de X <1;X> de A=>B: <A;B>
- Toutes les fonctions même stables ne correspondent pas à des termes.
- Mais l'absence de certaines fonctions stables montre l'absence de preuves/fonctions syntaxiques.

Utilisation en sémantique

- Pourrait être utilisé pour éviter de composer, car ce sont des modèles modulo beta réduction.
- Eléments de sens compatibles, composés, etc.: excitant, mais à ce jour, peu d'utilsiation.
- Ontologies: Fouqueré, Abrusci
- Sémantique compositionnelle lexicale, en utilisant la structure des types (cf. 2^e partie).

Second ordre

- Foncteur
  - espace cohérent X -> espace cohérent T(X) construit avec => et &.
  - Inclusion/renommage g -> inclusion renommage T(g)
- Représentable! C'est l'espace cohérent $\bigwedge X\ T(X)$

AVEC PREMIER ORDRE: HYPERDOCTRINES (LAWVERE)

- Compliqué…
- Combiner la catégorie des preuves propositionnelles avec une catégorie représentant les termes.

A-T-ON VRAIMENT BESOIN DE LA QUANTIFICATION?

- Tout linguiste connaît une langue africaine.
- 2 visions (question scholastique), 2 preuves:
  - Variable, élément générique,… qui connaît nécessairement une langue africaine.
  - Liste des linguistes, pour chacun on fait la preuve qu'il connaît une langue africaine.
  - Domaine connu, on peut imaginer une preuve différente pour chaque linguiste:
    le pour tout est une conjonction.
  - Si le domaine est infini cela cesse d'être des preuves au sens usuels, car la largeur des preuves et donc la preuve peut être infinie (mais chaque branche reste de hauteur finie): c'est du genre ω-règle de Gentzen.

RÉSUMÉ

- Logique classique:
  - Modèles simples de la prouvabilité, avec complétude, pour le premier ordre et pour les suivants aussi.
  - Toutes les preuves sont équivalentes, pas de modèles de preuves.
- Logique intuitionniste / lambda calcul typé
  - Modèles de la prouvabilité plus complexes, pour le premier ordre et pour les suivants aussi, avec résultats de complétude.
  - Modèles catégoriques des preuves et termes
    - Seulement pour la logique intuitionniste
    - Propositionnel et second ordre propositionnel
    - Plus problématique pour le premier ordre.
    - Possiblement combinable avec les modèles de la prouvabilité.
- Logique linéaire (théorie des types similaire à celle de la logique intuitionniste, mais plus raffinée)
  voir fin de la 2e partie pour son utilisation en sémantique.

2$^{\mathrm{e}}$ partie:

Towards a logical model of

some aspects of lexical semantics

joint work with Bruno Mery and Christian Bassac

Lexical Semantics within Compositional Semantics

- A not-so-recent problem: polysemy
- Sense disambiguation and lexical semantics
- Linguistic and background knowledge
- The advent of the Generative Lexicon
- A gap within the formalism

Typical examples from Pustejovsky's Generative Lexicon

- Qualia
  - *A quick cigarette* (telic)
  - *A partisan article* (agentive)
- Dot Objects
  - *An interesting book* (I)
  - *A heavy book* (φ)
  - *A large city* (T)
  - *A cosmopolitan city* (P)
- Co-predications
  - *A heavy, yet interesting book*
  - *Paris is a large, cosmopolitan city*
  - ? *A fast, delicious salmon*
  - ?? *Washington is a small city and signed a trade agreement with Paris*

Back to the roots: Montague semantics. Types.

Simply typed lambda terms $types ::= e \mid t \mid types \rightarrow types$

chair , *sleep* $e \rightarrow t$

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

Back to the roots: Montague semantics. Syntax/semantics.

(Syntactic type)*	$=$	Semantic type	
S^*	$=$	t	a sentence is a proposition
np^*	$=$	e	a noun phrase is an entity
n^*	$=$	$e \rightarrow t$	a noun is a subset of the set of entities
...	$=$	...	extends easily to all syntactic categories when a CG is used

Back to the roots: Montague semantics. Logic within lambda-calculus 1/2.

Logical operations (and, or, some, all the,.....) need constants:

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

Back to the roots: Montague semantics. Logic within lambda-calculus 2/2.

Words in the lexicon need constants for their denotation:

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

Back to the roots: 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 its normal form corresponds to a formula, the ”meaning”.

Back to the roots: Montague semantics. Computing the semantics. 2/5

word	***semantic type*** u^* ***semantics :*** λ***-term of type*** u^* x_v ***means that the variable or constant*** x ***is of type*** v
some	$(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$ $\lambda P_{e\rightarrow t}\ \lambda Q_{e\rightarrow t}\ (\exists_{(e\rightarrow t)\rightarrow t}\ (\lambda x_e(\wedge_{t\rightarrow(t\rightarrow t)}(P\ x)(Q\ x))))$
statements	$e \rightarrow t$ $\lambda x_e(\mathtt{statement}_{e\rightarrow t}\ x)$
speak_about	$e \rightarrow (e \rightarrow t)$ $\lambda y_e\ \lambda x_e\ ((\mathtt{speak\_about}_{e\rightarrow(e\rightarrow t)}\ x)y)$
themselves	$(e \rightarrow (e \rightarrow t)) \rightarrow (e \rightarrow t)$ $\lambda P_{e\rightarrow(e\rightarrow t)}\ \lambda x_e\ ((P\ x)x)$

Back to the roots: 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) (themsleves speak_about)) of type t

Back to the roots: Montague semantics. Computing the semantics. 4/5

$$\Big(\big(\lambda P_{e\to t}\ \lambda Q_{e\to t}\ (\exists_{(e\to t)\to t}\ (\lambda x_e(\wedge(P\ x)(Q\ x)))))\big(\lambda x_e(\mathtt{statement}_{e\to t}\ x)\big)\Big)$$
$$\Big(\big(\lambda P_{e\to(e\to t)}\ \lambda x_e\ ((P\ x)x)\big)\big(\lambda y_e\ \lambda x_e\ ((\mathtt{speak\_about}_{e\to(e\to t)}\ x)y)\big)\Big)$$

$$\downarrow\beta$$

$$\big(\lambda Q_{e\to t}\ (\exists_{(e\to t)\to t}\ (\lambda x_e(\wedge_{t\to(t\to t)}(\mathtt{statement}_{e\to t}\ x)(Q\ x))))\big)$$
$$\big(\lambda x_e\ ((\mathtt{speak\_about}_{e\to(e\to t)}\ x)x)\big)$$

$$\downarrow\beta$$

$$\big(\exists_{(e\to t)\to t}\ (\lambda x_e(\wedge(\mathtt{statement}_{e\to t}\ x)((\mathtt{speak\_about}_{e\to(e\to t)}\ x)x)))\big)$$

Back to the roots: Montague semantics. Computing the semantics. 5/5

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

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

This is the semantics of the analyzed sentence.

More general types and terms. Many sorted logic. TY_n

Extension to TY_n without difficulty nor suprise: e can be divided in several kind of entities (a kind of a flat ontology).

More general types and terms. Second order types.

One can also add type variables and quantification over types.

- Constants e and t, as well as any type variable α 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.

More general types and terms. Second order terms.

- A variable of type T i.e. $x:T$ or x^T is a *term*. [For each type, a denumerable set of variables of this type.]
- $(f\ \tau)$ is a term of type U whenever $\tau:T$ and $f:T\rightarrow U$.
- $\lambda x^T.\ \tau$ is a term of type $T\rightarrow U$ 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, .

More general types and terms. Second order 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).

More general types and terms. A second order example.

How to coordinate over any type of entities

predicates $P^{\alpha\to t}$ and $Q^{\beta\to t}$ over entities of respective kinds α and β

when we have a morphism from any type ξ to α and one from ξ to β?

$\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to a}\lambda g^{\xi\to b}.(\mathrm{and}\ (P\ (f\ x))(Q\ (g\ x)))$

One can even quantify over the predicates P,Q and the types α,β to which they apply:

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

Principles of our lexicon

- Remain within reach of Montagovian compositional semantics
- Allow both predicate and argument to contribute lexical information to the compound
- Integrate within existing discourse models

We advocate a system based on *optional modifiers*.

Overview of the Lexicon

How much information should a lexicon store ?

- Basic compositional data: number, type, optional character of arguments
- Lexical data for adaptations: qualia, dot objects. . .
- Constraints on modifiers induced by lexical data
- Interpretation(s) of each term

The Types

- Montagovian composition:
  - Predicate include the typing and the order of its arguments.
- Generative Lexicon style concept hierarchy:
  - Types are different for every distinct lexical behavior
  - A kind of ontology details the specialization relations between types
  - The result is close to a language-independent hierarchy of concepts

Second-order typing, like Girard's F system is needed for arbitrary modifiers:

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

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
  - $\Lambda\alpha\beta\lambda x^{\alpha}y^{\beta}f^{\alpha\to A}g^{\beta\to F}.((\mathrm{eat}^{A\to F\to t}\ (f\ x))\ (g\ y))$
  - Paris^{T}

The Terms: Optional Morphisms

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

A Complete Lexical Entry

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

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

Global vs local use of optional morphisms. GLOBAL

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.

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.

Global vs local use of optional morphisms. LOCAL

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

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

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

This can be done for all the occurrences of x and different $A,B,...$ and different $f,g,...$ can be used each time.

Second order typing is required to anticipate the yet unknown type of the argument and to factor the different types for f that will be use in the slots.

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

Standard behaviour

ϕ: physical objects

small stone

$$\overbrace{(\lambda x^{\varphi}.\ (\mathrm{small}^{\varphi\to\varphi} x))}^{\mathrm{small}}\ \overbrace{\tau^{\varphi}}^{\mathrm{stone}}$$

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

Qualia exploitation

wondering, loving smile

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

Facets (dot-objects): incorrect copredication

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

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

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.

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

Conjunction of $cospl^{P\to t}$, $\mathrm{cap}^{T\to t}$ and $port^{L\to t}$, applied to tk^T

If $T = P = L = e$, (Montague)
$(\lambda x^e(\mathrm{and}^{t\to(t\to t)}((\mathrm{and}^{t\to(t\to t)}\ (\mathrm{cospl}\ x)\ (\mathrm{cap}\ x))\ (\mathrm{port}\ x)))\ k$.

AND between three predicates over different kinds $P^{\alpha\to t}$, $Q^{\beta\to t}$, $R^{\beta\to t}$

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

The morphisms f, g and h convert x to **different** types.

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

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

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

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

$$(\mathrm{and}^{t\to(t\to t)}\ (\mathrm{and}^{t\to(t\to t)}\ (\mathrm{cospl}\ (f_p\ k^T)^P)^t)\ (\mathrm{cap}\ (\mathrm{id}\ k^T)^T)^t)^t\ (\mathrm{port}\ (f_l\ k^T)^L)^t)^t$$

Importing an existing lexicon

- Main type and argument structure: main λ-term
- *Qualia*-roles: local modifiers
- Dot objects: local modifiers
- Some specific constructions are global modifiers (e.g. grinding).
- Inheritance structure: local modifier $\rightarrow$ parent

The calculus, summarized

- First-order λ-bindings: usual composition
- Open slots: generate all combinations of modifiers available
- As many interpretations as well-typed combinations

Paris is an populous city by the Seine river

$$((\Lambda\xi\ .\ \lambda x^{\xi} f^{\xi\to P} g^{\xi\to L}\ .\ (\mathrm{and}\ (\mathrm{populous}^{P\to t}\ (f\ x))\ (\mathrm{riverside}^{L\to t}\ (g\ x))))$$

$$\{T\}\ \mathrm{Paris}^{T}\ \lambda x^{T}\ (f_P^{T\to P}\ x)\ \lambda x^{T}\ .\ (f_L^{T\to L}\ x))$$

Logical Formulæ

- Many possible results
- Our choice: classical, higher-order predicate logic
- No modalities

$$\mathrm{and}(\mathrm{populous}(f_P(\mathrm{Paris}),\ \mathrm{riverside}(f_L(\mathrm{Paris})))$$

Intermezzo: my favorite puzzle. Situation.

A shelf.

Three copies of *Madame Bovary*.

The collected novels of Flaubert in one volume (L'éducation sentimentale, Madame Bovary, Bouvard et Pécuchet)

One copy of *Jacques le fataliste*.

The volume also contains *Trois contes: Un coeur simple, La légende de Saint-Julien, Salammbô*

Intermezzo: my favorite puzzle. Questions.

- I carried down all the books to the cellar.
- Indeed, I read them all.
- How many books did you carry?
- How many books did you read?

Critics

- The classical solution with products: $\langle p_1(u), p_2(u)\rangle = u$

- (Asher's solution with pullbacks) too tight relation type structure / morphisms (only and always canonical morphisms) and unavoidable relation to product

- (Ours) not enough relation types/morphisms (no relation at all), typing does not constrain morphims,

Linear alternative

Direct representation with monoidal product $A \otimes B$ and replication !

- $A \otimes B$
  - without $\langle p_1(u), p_2(u)\rangle = u$
  - without canonical morphism
  - but the type of a transformation relates to the structure of the type.

- Types of morphims in a linear setting ($\vdash$ being $\multimap$) either:
  - <u>irreversible:</u> $A \multimap U$ since $A \not\multimap U \otimes A$
  - <u>reusable:</u> $A \rightarrow B = (!A) \multimap U$ since $(!A) \multimap U \otimes (!A)$

This leads to general questions....

Which logic for semantics? Linear Logic?

Two kind of logics:

- glue language?
  - usually base types e,t constructor $\rightarrow$
  - not rich enough
  - composition better handled with linear types
- language of semantic representation
  - usually undefined, fragment of Higher Order Logic
  - too rich, but not enough fine grained enough. Linear logic?

A natural representation (too natural?)

In the usual system we use the following: if the lambda constants are connectives, quantifiers and relational or functional symbols, then every closed temr of type t is a formula, etc.

What about a closed term of type $e \to (e \otimes t)$ and other complex types.

Interpretation, models

- usually possible worlds
- too large, uncomputable, ...
- no well defined, unless free or categorical semantics
- can we use models of linear logic (of formulae or of composition)

Argument for and against linear logic. For.

For:

- Refined both for semantic representation and as a description of the computation leading to these representations.

- Encode usual formulae and even usual typed lambda calculus.

Argument for and against linear logic. Against.

Against

- As opposed to usual semantics, no good model of first order. Phase valued models unatural, *ad hoc*
- Models of composition computation cannnot handle proper axioms — coherence spaces (Scott domains), ludics (game semantics)

[Both are even needed for maths, hence for linguistics...]

A direction that I am exploring (me but also Melliès, Lamarche,) refinement **sheaves models** of intuistionistic logic (topoi, local notion — Grothendieck, Lawvere, Lambek)

Yet more general questions. 1

performance / competence

cognitive experiments versus formal computational complexity

Algorithmic complexity not adapted. Logical complexity:

- $\rightarrow$ nesting $(e \rightarrow t) \rightarrow t$
- quantifier alternation
- order (individuals, predicates, predicates of predicates,...)

Yet more general questions. 2

How things are and works / How a specific language describes this

Ambiguity: does the lexicon (e.g. qualia structure) describe

- the world of the discourse universe (ontology)
- or a language dependent ontology:

  Ma voiture est crevée. even *J'ai crevé. (une roue de ma voiture est crevée).*
  ** Ma voiture est bouchée. (le carburateur)* or ** Ma voiture est à plat. (la batterie)*

Such examples as well as cross linguistic comparisons indicate a distinction should be made.