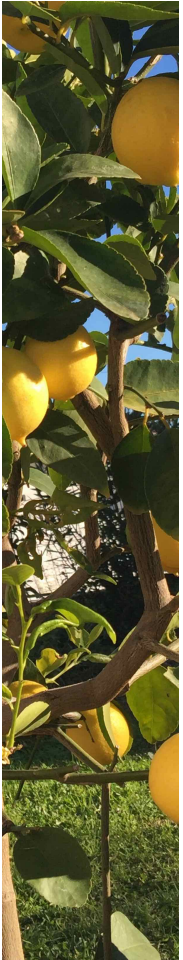
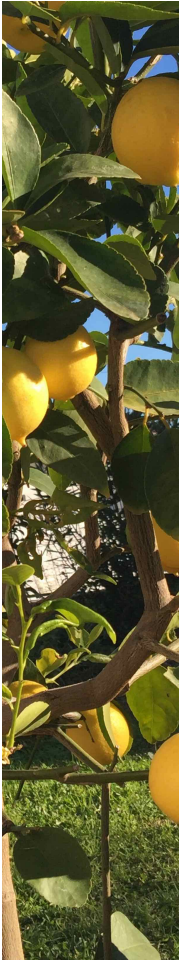


Compréhension automatique du langage naturel et théories des types

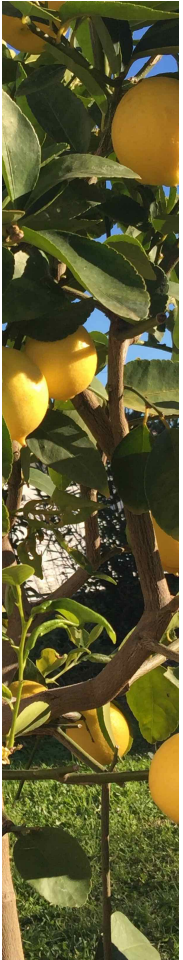
Christian Retoré

Équipe Texte, LIRMM, Université de Montpellier

INRIA-Sophia, 25 janvier 2019



A Traitement automatique des langues



A.1. Famille de problèmes à traiter

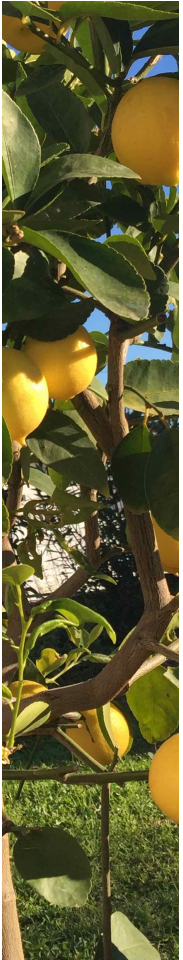
Deux tâches classiques

A analyse automatique (de texte, de parole, annoté ou non) résultat ?
B génération automatique (de texte, de parole) à partir de quoi ?

Traduction automatique : 2nde guerre mondiale.

Domaine propre (recherche d'information) mais aussi utile pour A et B ci-dessus

— fouille, acquisition (des données notamment linguistique pour les autre tâches ou RI)



A.2. Applications

Traduction automatique, aide à la traduction

> The flesh is weak but the spirit is willing.
> The meat is rotten but the vodka is strong.

Aide à la traduction : domaine spécifique + interaction
Correcteurs orthographiques (Word(Synapse) : vert)

(1) QuelS livreS crois-tu qu'il sait que je pense que tu as luS ?

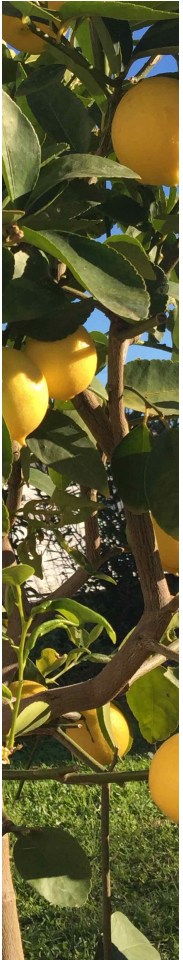
Dialogue homme machine

(2) Les enfants prendront une pizza.

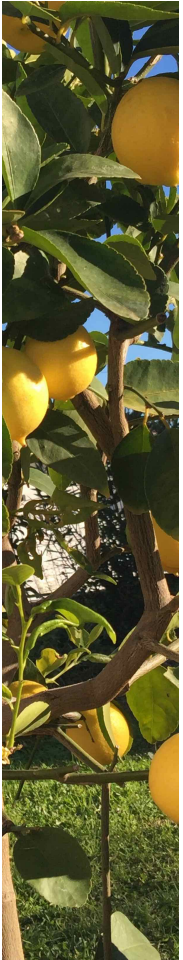
Résumé automatique seule la méthode bête marche
Réponse à des questions (question answering cf. text entailment)

> Geach était il l'élève de Wittgenstein ?

Domaine relié : **recherche d'information** Big différence : Big Data, analyse superficielle, méthodes statistiques, machine learning



B Les niveaux d'analyse de la langue & leurs méthodes



B.1. Les sons : phonétique/phonologie

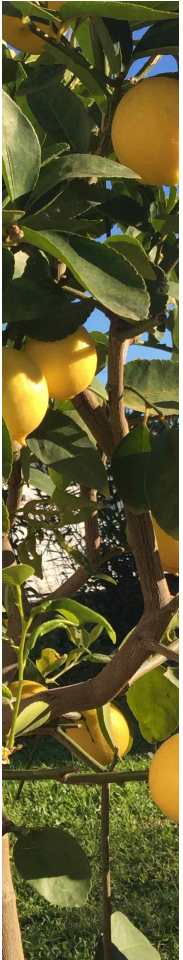
B.1.a Phonétique

Acoustique / système phonatoire/auditif Traitement du signal / médecine

B.1.b Phonologie

Les sons abstraits : système discret (automates / transducteurs)

(3) Bali / Paris indistincts pour un japonais



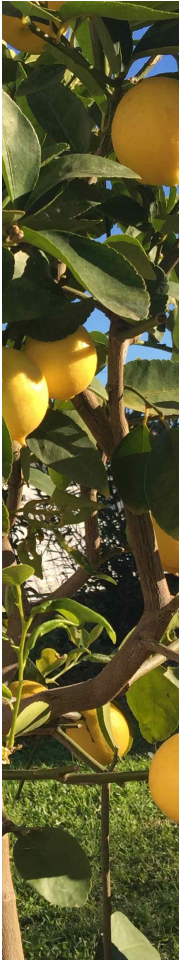
B.2. Les mots : morphologie

B.2.a Morphologie dérivationnelle

changement de catégorie possible

allumer/allumage
témoigner/témoignage
maquiller/maquillage
garer/garage

maison /maisonnette
camion /camionnette
carpe/carpette



B.2.b Morphologie flexionnelle

pas de changement de catégorie (automates et transducteurs)

cheval $\rightarrow$ chevaux

aller $\rightarrow$ allons

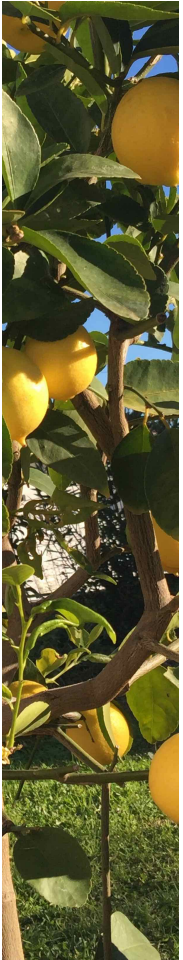
aller $\rightarrow$ va

aller $\rightarrow$ irons

B.2.c Étiquetage grammatical

nom, verbe, déterminant etc.

Automates ou probas (modèles de Markov cachés)



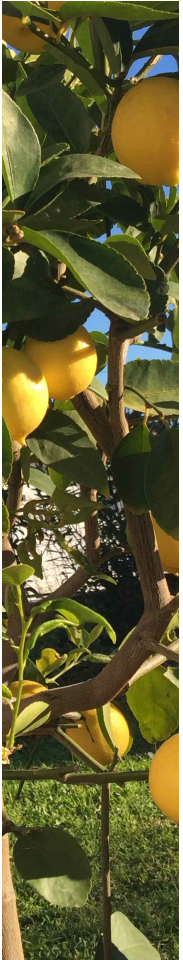
B.3. Analyse de la phrase (arbre) : syntaxe

(4) Réparer fais les la le.
(5) *Je fais la réparer
(6) Je la fais réparer
(7) * [[Pierre [mange une]] pomme]
(8) Pierre [mange [une pomme]]

Grammaires formelles (CFG, TAG etc)

Déductions formelles dans une logique théorie des types sous structurelle (cf. ci-après)

Arbres + contraintes (par ex. "le sujet précède le verbe")



B.4. Le sens : sémantique

B.4.a Le sens d'un mot : sémantique lexicale

Sens, restrictions de sélection, mots associés (logique, probabilités, graphes (petit monde), jeux)

Rôle télique : être lu, informer, cultiver,

Rôle constitutif : pages, couvertures

Rôle agentif : imprimeur, auteur,

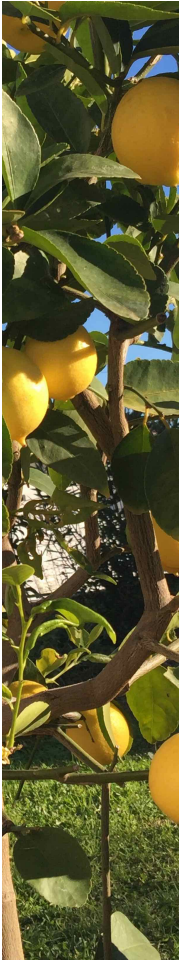
B.4.b Le sens d'une phrase : semantique compositionnelle, formelle, logique

Phrase -> formule logique (modale, ordre supérieur, théorie des types)

Sens : mondes possibles dans lequel la phrase est vraie

Lambda calcul simplement typé : représente et calcule les formules par compositionnalité.

Approches différentes mais compatibles.

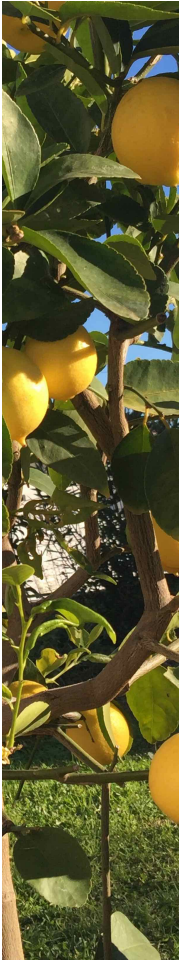


B.5. L'interprétation en en contexte (discours, dialogue) : pragmatique

Prise en compte du contexte linguistique et extra linguistique

(9) a. Il est tombé. Quelqu'un l'a poussé.
b. "l'"="il" + causalité

(10) a. Allons plutôt dans ce restaurant.
b. "nous" ? "ce" ?



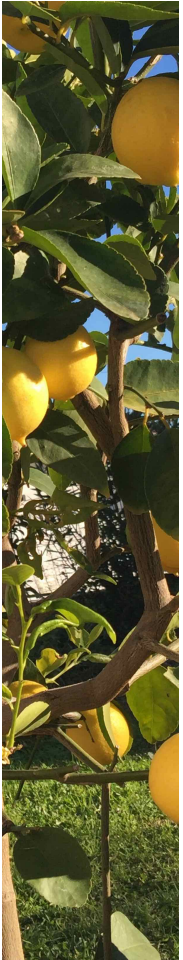
B.6. Deux notions de sémantique

De quoi ça parle (analyse de texte, par des méthodes statistiques).

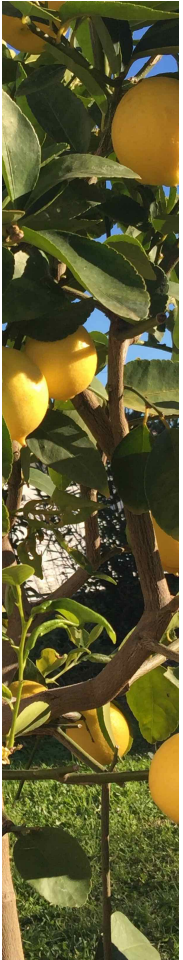
Qui fait quoi, ce qui est affirmé et réfuté (analyse de quelques phrases par des méthodes logiques).

> Geach était il l'élève de Wittgenstein ?
>
> Wikipédia :
>
> En 1941, IL épousa la philosophe Elizabeth Anscombe, grâce à LAQUELLE IL entra en contact avec Ludwig Wittgenstein. Bien qu'IL N'ait JAMAIS suivi l'enseignement académique de CE DERNIER, cependant IL EN éprouva fortement l'influence.

Dans cet exposé : syntaxe (déductive) et sémantique (logique) de la phrase et des mots.



C Analyse logique du sens d'une phrase



C.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

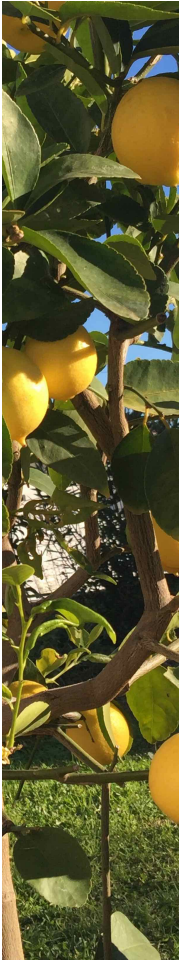
au moins deux hypothèses

A hypothèse la plus à gauche

$$\dots [A] \dots \dots$$
$$\vdots$$
$$\frac{B}{A\backslash B}\ \backslash_i$$

— l'hyp. A est annulée

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



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

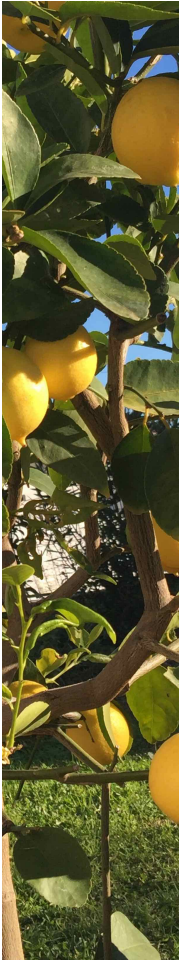
au moins deux hyp. libres

A hyp. libre la plus à droite

$$\dots\dots[A]\dots$$
$$\vdots$$
$$\frac{B}{B/A}\,/_i$$

— l'hyp. A est annulée

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

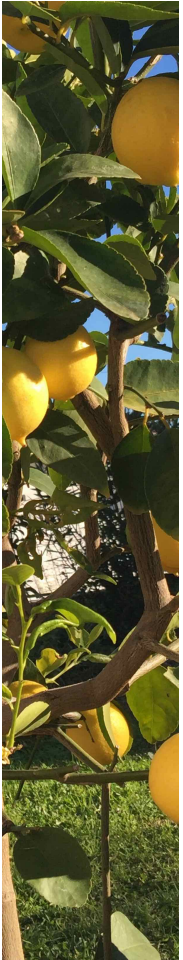


C.3. Analyse syntaxique = déduction (parsing as deduction)

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 :

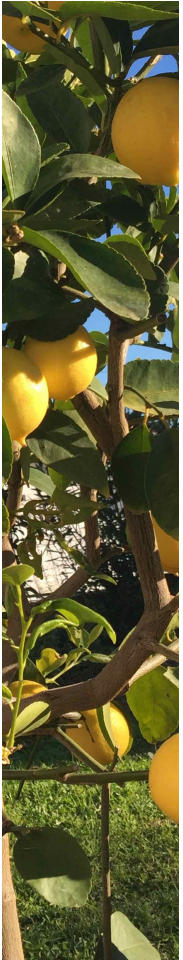
$$\forall i \exists c_i \in Lex(m_i) \quad c_1 \dots c_n \vdash S$$



C.4. Sémantique compositionnelle : principes

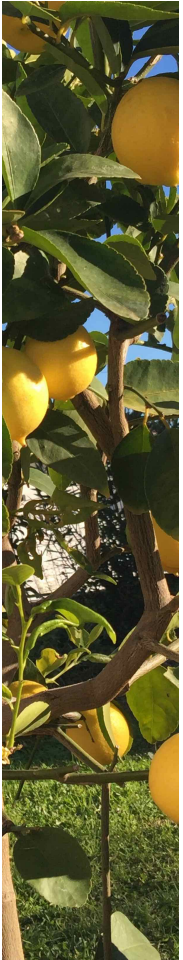
Le sens d'une expression composée est fonction du sens de ses parties (Frege) et de leur assemblage syntaxique (Montague)

$(\text{Catégorie syntaxique})^*$	$=$	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



C.5. Des constantes pour les opérations logiques

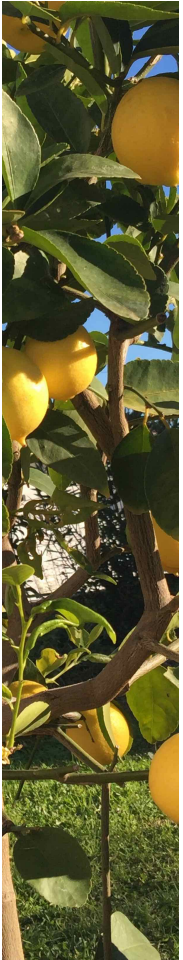
Constant	Type
$\exists$	$(e \to t) \to t$
$\forall$	$(e \to t) \to t$
$\wedge$	$t \to (t \to t)$
$\vee$	$t \to (t \to t)$
$\supset$	$t \to (t \to t)$



C.6. Des constantes pour les prédicats du langage

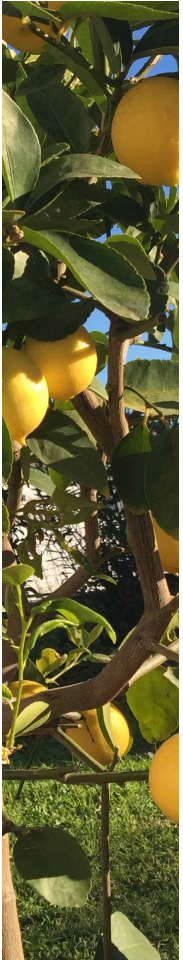
La dénotation des mots requiert des prédicats :

aime	$\lambda x \lambda y\ (\text{aime}\ y)\ x$	$x : e,\ y : e, \text{aime} : e \to (e \to t)$
« aime » est un prédicat binaire		
Garance	$\lambda P\ (P\ \text{Garance})$	$P : e \to t,\ \text{Garance} : e$
« Garance » est décrite comme les propriétés de « Garance »		



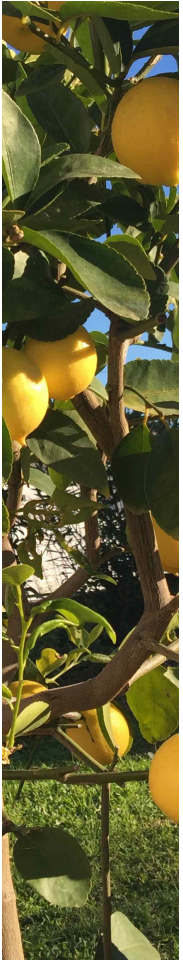
C.7. Sémantique à la Montague : algorithme

1. analyse syntaxique preuve de S
2. conversion en lambda terme de type t sur e et t
3. insertion des lambda terme lexicaux (même type)
4. réduction
   $\rightarrow$ terme de type t
   = formule logique
   = sens de la phrase analysée



C.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	$(np\backslash S)/np$ $e \to (e \to t)$ $\lambda y^e\ \lambda x^e\ ((\texttt{prendront}^{e\to(e\to t)}\ x)y)$

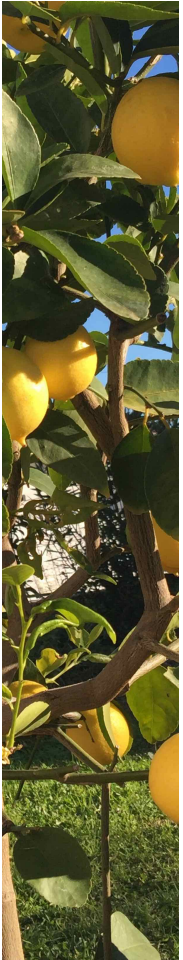


C.9. Analyse syntaxique ∃∀

Il y a deux analyses syntaxiques possibles. Une :

∃∀

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



C.10. Syntaxe $\rightarrow \lambda$-terme sémantique de la phrase

$\exists\forall$

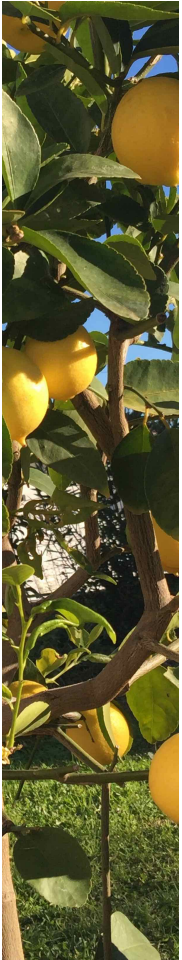
$$\dfrac{\dfrac{\dfrac{\dfrac{\textit{les}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}} \quad \dfrac{\textit{enfants}}{(\mathbf{e}\rightarrow\mathbf{t})}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}}\rightarrow_e \quad \dfrac{\dfrac{\textit{prendront}}{\mathbf{e}\rightarrow\mathbf{e}\rightarrow\m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

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

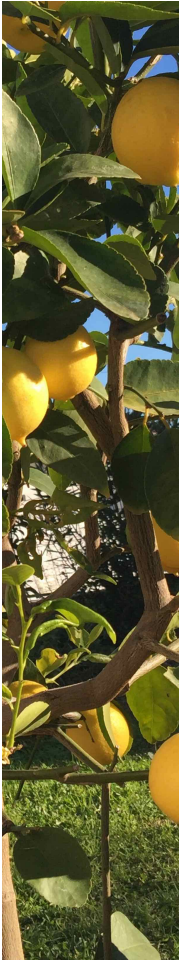


C.11. Calculs, par étapes 1/2

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

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

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

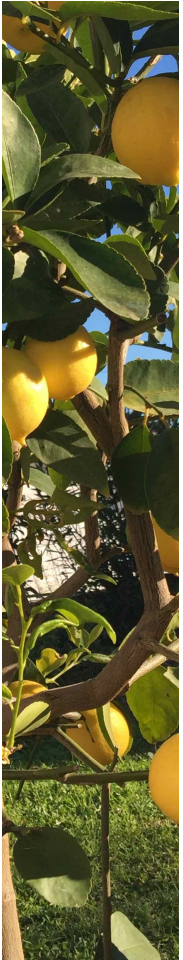


C.12. Calculs, par étapes 2/2

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

ce qui s'écrit communément :

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



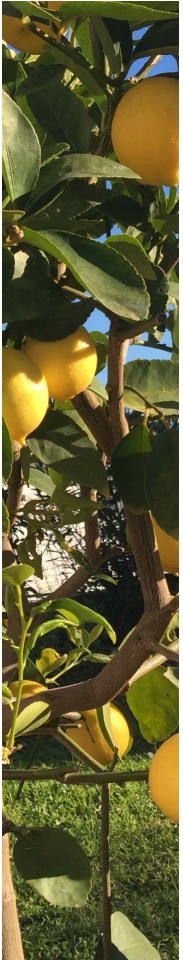
C.13. Avec l'autre analyse syntaxique...

∀∃

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

Qui correspond à l'analyse :

∀∃



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

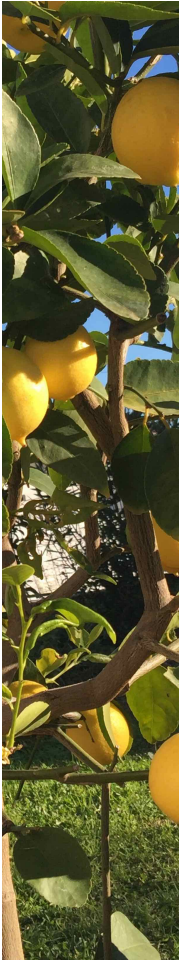
$$\dfrac{\dfrac{\mathbf{(e\to t)\to(e\to t)\to t} \quad \mathbf{(e\to t)}}{\mathbf{(e\to t)\to t}}\to_e \qquad \dfrac{\mathbf{t}}{\mathbf{e\to t}}\to_i(1)}{\mathbf{t}}\to_e$$

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)



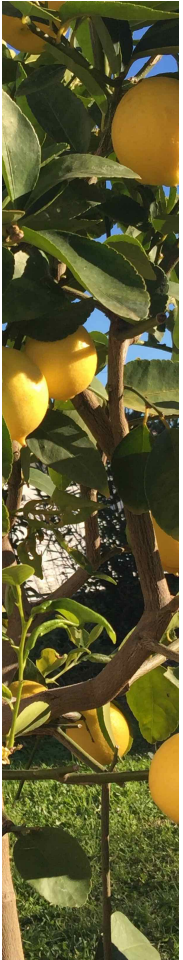
C.14. Calculs (bis repetita placent)

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

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

ce qui s'écrit communément :

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

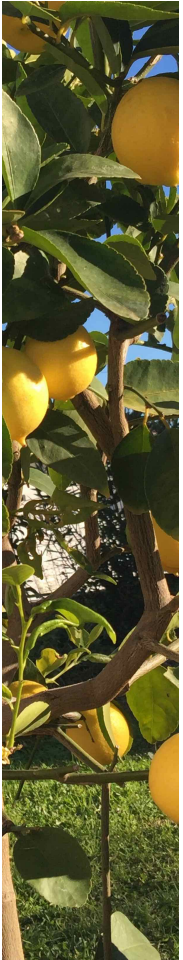


C.15. Comment construire un lexique sémantique ?

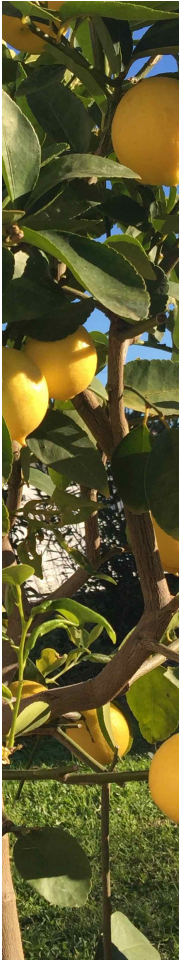
Acquisition des λ-termes lexicaux : pas facile... même avec de bonne annotations.

Mots grammaticaux (prépositions, conjonctions, pronoms, etc.) à la main.

Noms communs "cha" -> $\lambda x^{\mathbf{e}}.chat(x)$



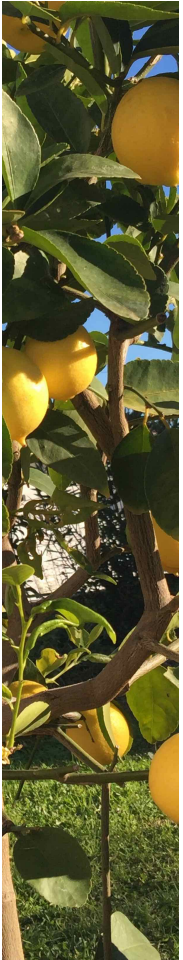
D Le sens des mots en contexte



D.1. Examples of Lexical Issues

Short roadmap :

- Basics : selection, polysemy, felicity
- Our system – a brief recap
- Classical GL constructions
- Co-predication and constraints
- Deverbals
- The narrative of travel
- Outstanding issues



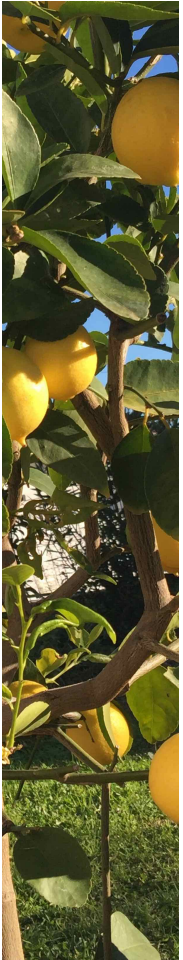
D.2. Restriction of Selection and Polysemy

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.

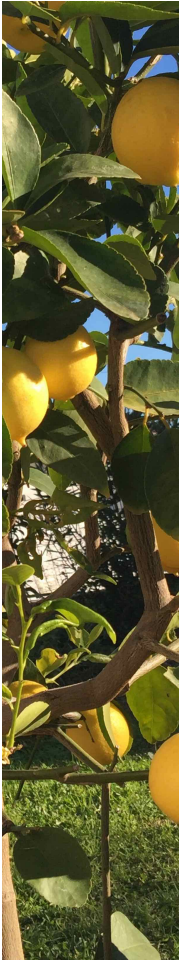


Contrastive ambiguity

— Different lexemes, homophonic / homographic
— Bank, Bar, Pen. . .

Relational polysemy

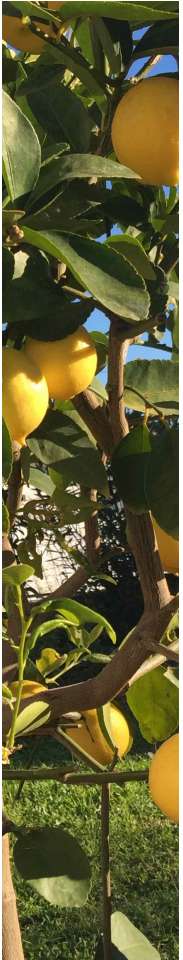
— A single word, different uses and related meanings
— The bank killed my account.
— My car was written out.
— The school is on strike.



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

Montague : everything is acceptable — All syntactically valid items have a meaning
- — All entities (logic individuals) the same semantic “nature”
- — We have to rely on pragmatics or interpretation

Strong restriction from lexical semantics
- — The type of entities $\mathbf{e}$ is replaced by many sorts
- — Barking dogs are licensed, (almost) everything else is blocked
- — No language works that way



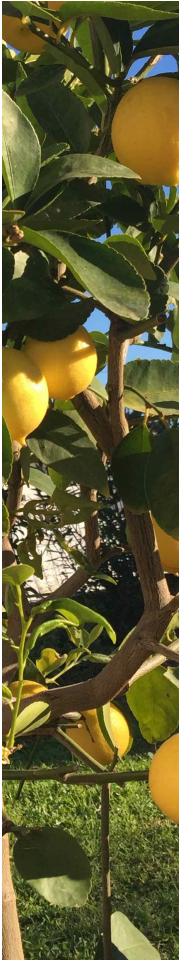
D.4. Acceptability and Felicity : Semantics, or Pragmatics ?

Creative uses and semantic licenses

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

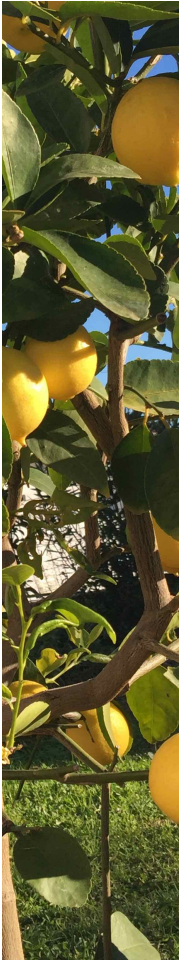
Pragmatic licenses

- Variations from the lexicon in specific contexts external to the utterances
- *I deceased again last evening*
- The analyser cannot be expected to get a correct meaning without the external context



D.5. A moderate solution

— Creative use should be included
— Anything that can be comprehended in a self-contained text should be included
— We should not try to account for unknown contextual information
— World knowledge, background, social and universal contexts can be integrated
— *Pragmatics* should not be a pretext to give up on critical variations of meaning



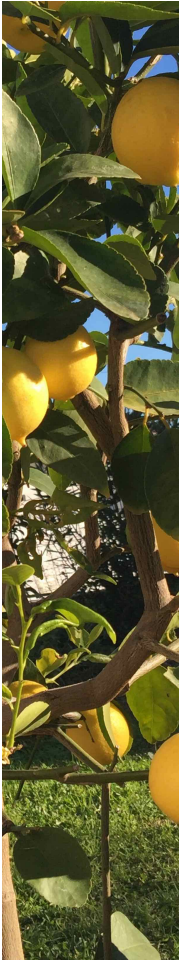
D.6. In a Nutshell

The Generative Lexicon

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

Framework sketch

- — 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 *modifiers*
- — Normal application is the same
- — Modifiers are used when types *clash*
- — Types *guide* the selection of modifiers



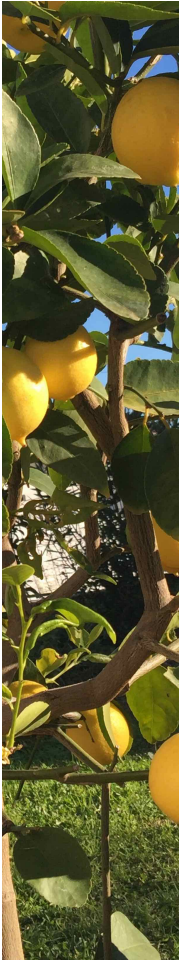
D.7. A lexicon

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 a huge place*
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_a_huge_place	$huge\_place : Pl \to \mathbf{t}$	
voted	$voted : P \to \mathbf{t}$	



1. Type mismatch in $huge\_place^{Pl\to\mathbf{t}}(Birmingham^{T}))$, resolved using t_3 :

$$huge\_place^{Pl\to\mathbf{t}}(t_3^{T\to Pl}Birmingham^{T}))$$

2. The same, using t_2 :

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

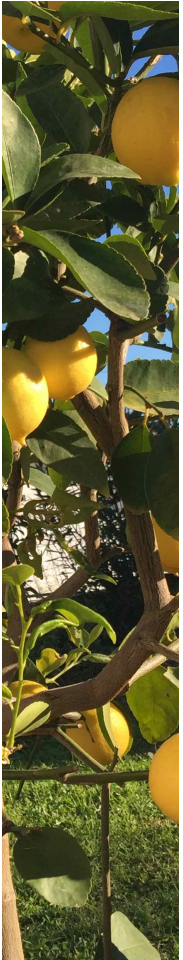
3. We use a polymorphic conjunction operator, $\&^{\Pi}$.
   Details later.

   $\Lambda\xi\lambda x^{\xi}\ \lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}(\mathrm{and}^{(\mathbf{t}\to\mathbf{t})\to\mathbf{t}}\ (huge\_place\ (f\ x))(voted\ (g\ x)))$

   After application, we have :

$$(\mathrm{and}\ (huge\_place^{Pl\to\mathbf{t}}\ (t_3^{T\to Pl}\ Birmingham^{T}))$$

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



D.8. Qualia Exploitation

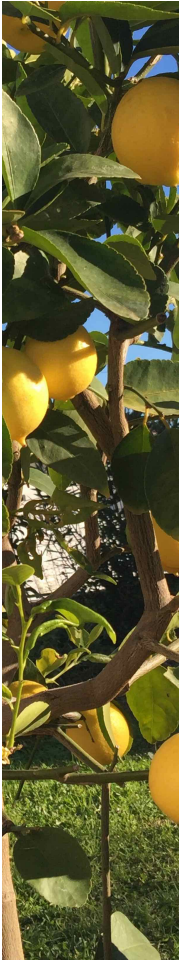
Classic GL : The "actual" meaning is conveyed by *qualia*

Formal An heavy sword (physical property)

Constitutive A well-tempered honed sword (blade, edge)

Agentive A master's sword (smith)

Telic An effective sword (for battle, practice, fruit-cutting. . .)



For instance, *books* are *written* (agentive) and *read* (telic) We have (on top of other things) transformations

$$f^{Book\to \mathbf{v}}_{Agentive}, f^{Book\to \mathbf{v}}_{Telic}$$

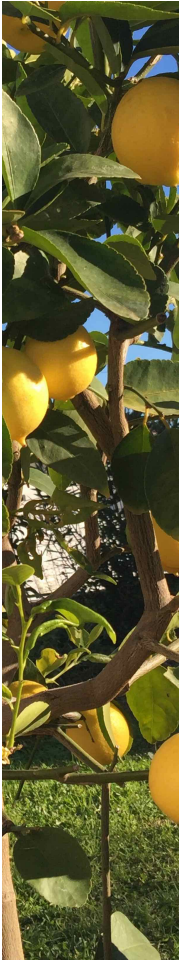
So *I finished the book* is ambiguous between

$$(\text{finished}^{A\to \mathbf{v}\to \mathbf{t}}\ I^{A}\ (f_{Agentive}\ \text{the\_book}))$$

and

$$(\text{finished}^{A\to \mathbf{v}\to \mathbf{t}}\ I^{A}\ (f_{Telic}\ \text{the\_book}))$$

(This is expected behaviour in GL-esque theories.)



D.9. Dot Objects

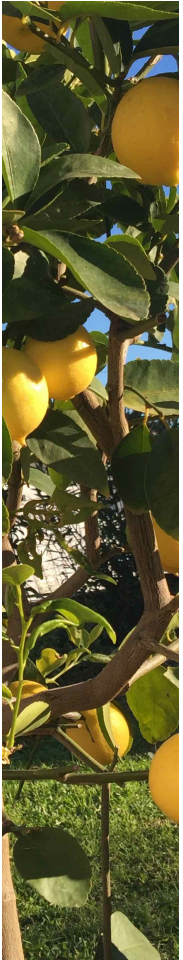
"Dots" are very special objects in GL. Examples :

- *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. . .*

(All those are co-predicative, more on that in a minute.)

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.10. Conjunction for co-predication

! Polymorphic conjunction :

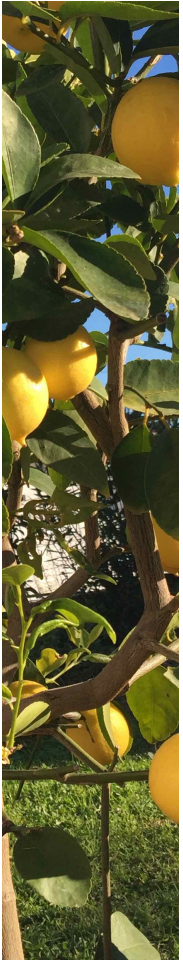
Given predicates $P^{\alpha\to\mathbf{t}}$, $Q^{\beta\to\mathbf{t}}$ over entities of respective types α, β,

given any type ξ with two morphisms from ξ to α, 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

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



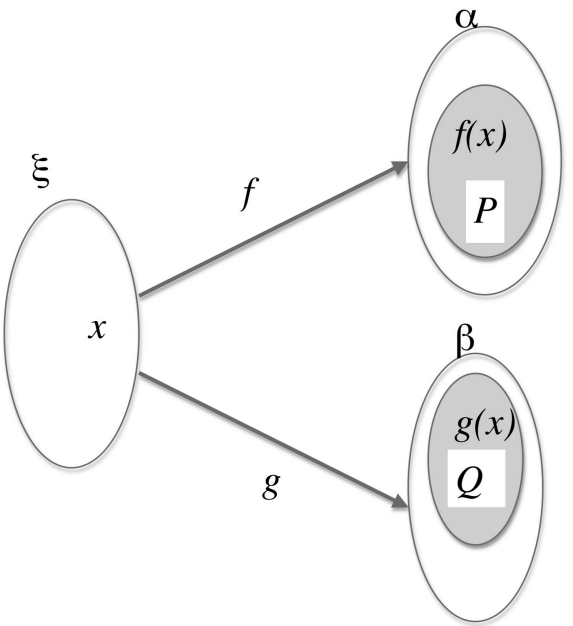
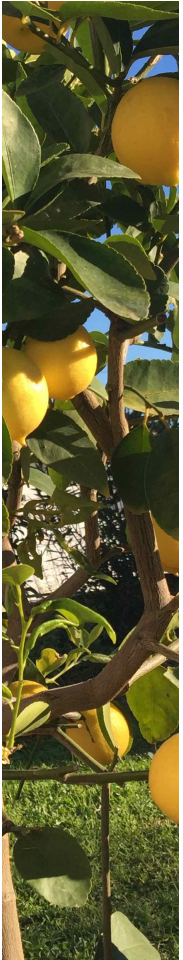


FIGURE 1 – Polymorphic conjunction : $P(f(x))\&Q(g(x))$ with $x : \xi$, $f : \xi \to \alpha$, $g : \xi \to \beta$.



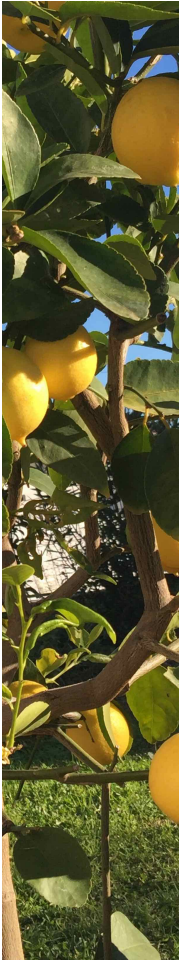
D.11. Constraints

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.
- Liverpool is a large city and voted labour.
- *Liverpool is a large city and won the cup.

Constraints on flexibility

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

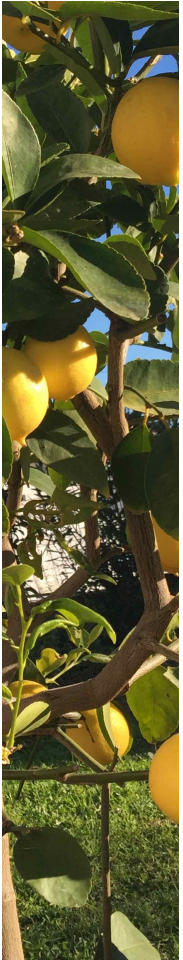


D.12. An example of a lexicon

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



D.13. An example of a derivation

Liverpool is a big place and voted

Polymorphic AND yields :

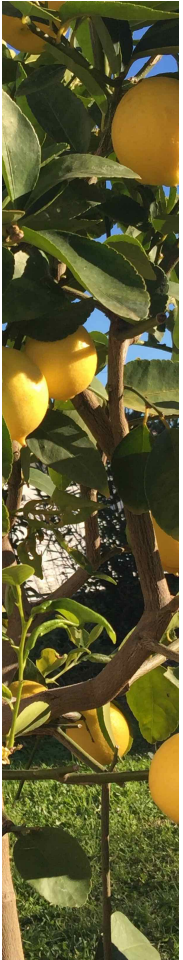
$$(\&^{\Pi}(big\_place)^{PI\to\mathbf{t}}(voted)^{P\to\mathbf{t}})$$

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

$$\&^{\Pi}\{PI\}\{P\}(big\_place)^{PI\to\mathbf{t}}(voted)^{P\to\mathbf{t}}$$

It reduces to :

$$\Lambda\xi\lambda x^{\xi}\ \lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}(\mathrm{and}^{\mathbf{t}\to\mathbf{t})\to\mathbf{t}}\ (big\_place\ (f\ x))(voted\ (g\ x)))$$

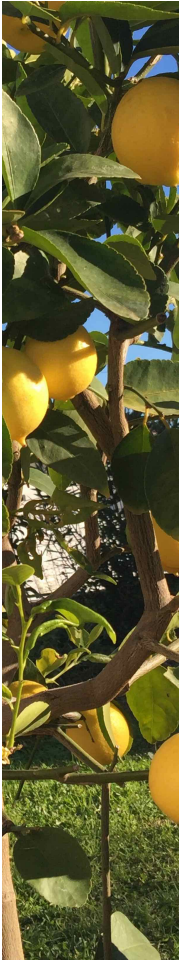


D.14. Unfortunately, more subtle than this

Syntactical relaxation of semi-flexible constraints

— *The salmon was lightning fast. It is delicious.*

— *Semi-flexible* : acts as *Rigid*, but is reset by reference.



D.15. Deverbal Nouns

(With Livi Réal)

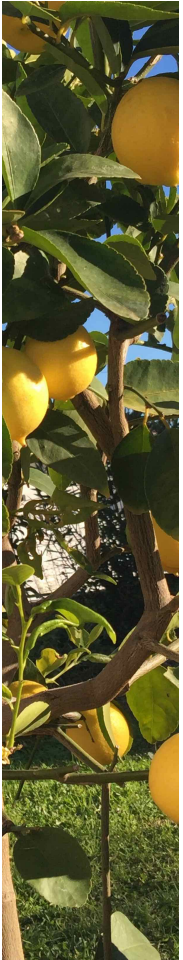
* *The construction was delayed and is around the corner.*

Long study to see what facets of deverbals are available and compatible with others

Specialised events, e. g. $\mathbf{v}_{\varphi}$. Example of *signature* :

- *The signature took a long time.* $\mathbf{v}$
- *The signature was an angry jab.* $\mathbf{v}_{\varphi}$
- *The signature doesn't stand out on the page.* φ

Our analysis : primary type $\mathbf{v}$, transformations $f^{\mathbf{v}\to\mathbf{v}_{\varphi}}, f^{\mathbf{v}\to\varphi}$



However, compare :

— * *The signature took three months and is in black ink.*
— * *The signature was hard to obtain and was neatly executed.*
— *The signature, that nearly pierced the page, is preserved in the National Library.*

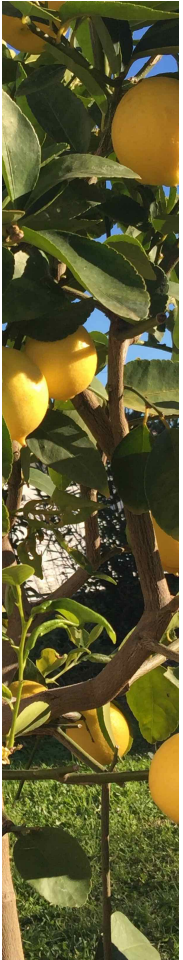
We get around this by having constraints : both transformations are flexible, but the *identity* is rigid

$$f^{\mathbf{v}\rightarrow\mathbf{v}_{\varphi}} : F$$

$$f^{\mathbf{v}\rightarrow\varphi} : F$$

$$id^{\mathbf{v}\rightarrow\mathbf{v}} : R$$

. . . So resultatives can have various facets *X-or* the original event



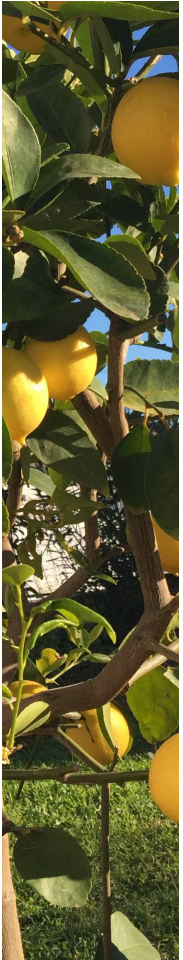
D.16. The Virtual Traveler

Project Itipy

- — Project by Région Aquitaine & UPPA
- — Reconstruction of itineraries from a corpus of travel stories
- — Restricted domain :
  - — Region : Pyrenean mountains
  - — Date : mainly XIX century
  - — Type : travel logs and stories

Some phrases

- — *. . . this road which climbs incessantly for two miles*
- — *. . . where the roads to Lux and to Pau branch off. The one to Lux enters a gorge which leads you to the bottom of a precipice and traverses the Gave de Pau*



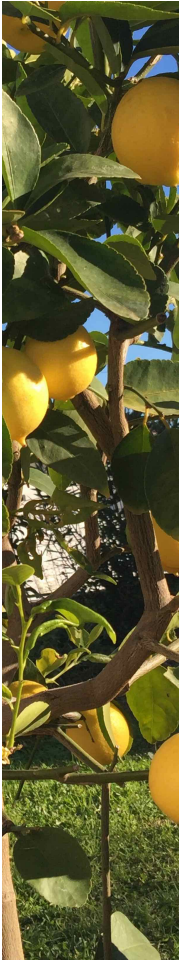
D.17. Solution to the Virtual Traveler

Preparations

— Wide-coverage grammar with a (single-sort) semantic lexicon
— Implementation in Grail with λ-DRSs

Augmented Lexical Semantics

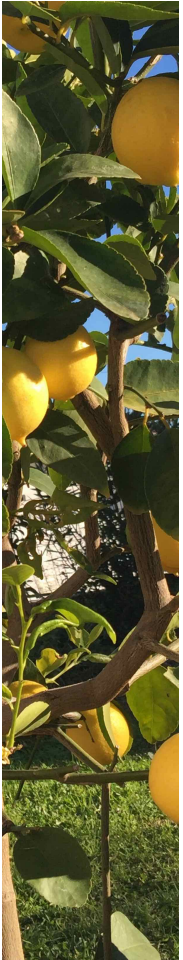
— Used for spatial data
— Base sorts : *Paths* and *Regions*
— *Regions* are divided according to a small taxonomy
— Transformation : *fictive motion*



D.18. Outstanding issues

Coming up are a few more mostly informal points :

- Framework comparisons
- The type system
- Better constraints
- Variations of lexica
- Implementation



D.19. Our guidelines vs other possibilities

Ontological Types

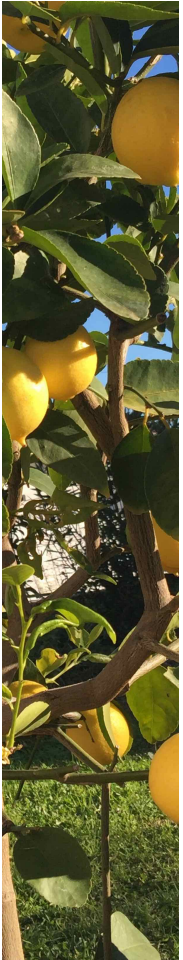
- — Most other approaches
- — Asher, Bekki, Pustejovski. . .
- — Ontology (concept) provides types
- — Types provide all adaptations (co-compositions, shifts, accommodations. . .)

Lexical Sorts

- — 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

Our approach

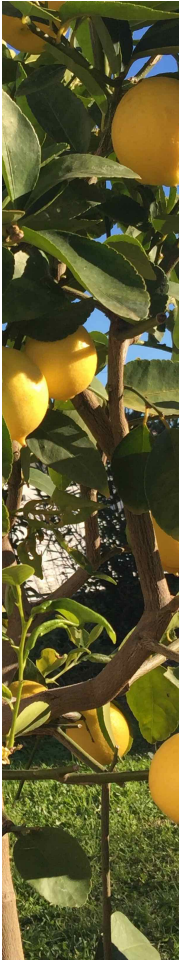
- — Based on the latter
- — General type-based transformations are still used
- — The lexicon can overload type transformations



— (Simply for reasons of scale and practicality)

Other possibilities

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

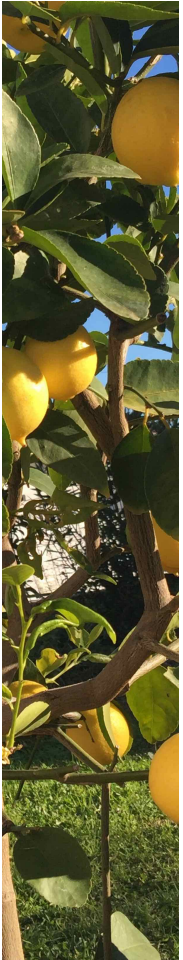


D.20. Issues of Granularity

— *How* should we build a (comprehensive) type system ?
— *What* would the sorts be ?
— *How big* shall we make it ?

Many different approaches :

1. Do not. (Keep **e** from Montague, not acceptable.)
2. Assume it exists. (Actually, most publications.)
3. Keep it restricted to a specific domain. (It works.)
4. Choose a dozen sorts or two. (*animated*, *physical*, *abstract*, *edible*. . .)
5. Every noun is a sort. (Luo.)
6. Every single-free-variable-formula can be a sort.



D.21. The Classifier Approach

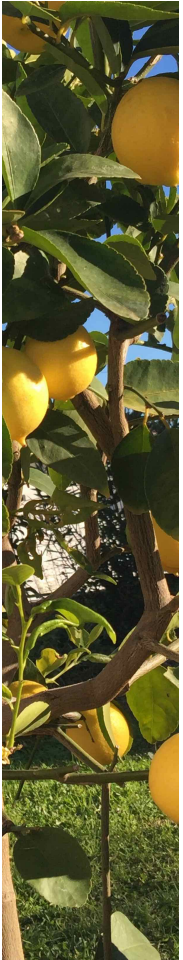
Features of the classifier systems

- Common to several language families
- Asian languages, Sign languages. . .
- Used for counting and measuring
- Apply to most entities in the lexicon

Granularity

- Many classifiers, but a lot less than the lexicon
- Semi-organized : generic, common, specialized
- Variations of use and lexical licenses
- Naturally occuring

Implicit sorts in English and French In progress



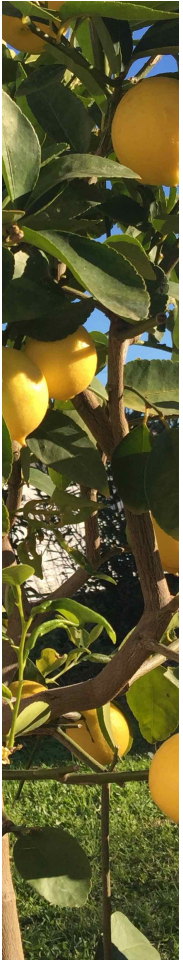
D.22. Linear Constraints

Current constraints

— Flexible (compatible with everything)
— Rigid (not compatible with anything)
— Plus syntax-based relaxation

Possible configurations

— Set A modifiers compatible with each others, but not with set B
— Using a modifier f_1 imply we may not use a modifier f_2, except if another g is used first
— . . . and even more complicated constraints
— *This is not a priority*, as we have no examples



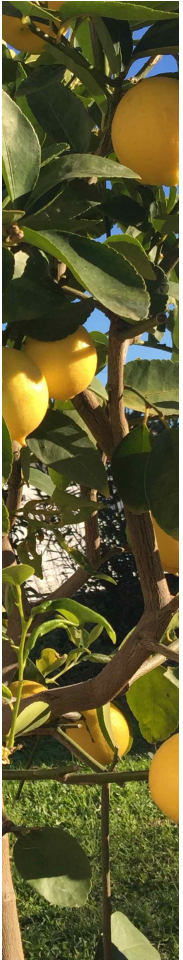
Linear Fromulation

— Idea : associate to lexemes a *linear formula* describing the available modifiers (rather than a list)

— If possible sets are $\{k\}, \{f, g\}, \{g, h\}, \{h, f\}$, excluding any other configurations. . .

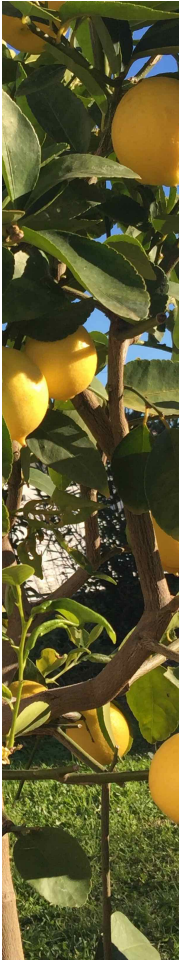
— That formula is : $!k\&(!f \otimes !g)\&(!g \otimes !h)\ \&(!h \otimes !f)$

— (The actual results are somewhat simpler than this, to appear.)



D.23. Felicity

- Infelicitous statements can receive a meaning from context
- In most applications, meaningfulness should be assumed
- (Exceptions include auto-completion and dictation !)
- When a type clash is not solvable :
  1. generate an appropriate modifier
  2. try to infer its properties



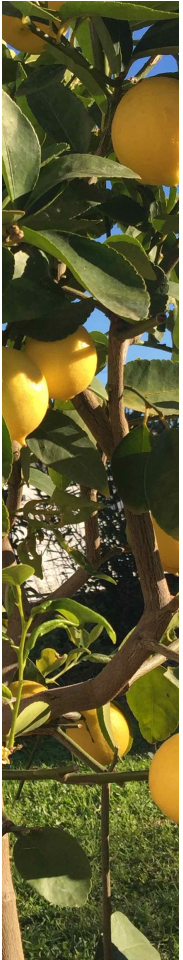
D.24. Strong Idiosyncrasy

Linguistic constructs are not all cognitively motivated
idioms and specific constructs are illustrations of this.

Differences of language
I have punctured

Differences of dialect
One half-strawberry

Differences of jargon
Straighten #16



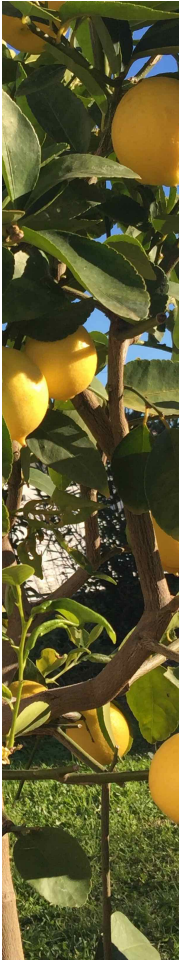
D.25. Contexts and Variations of Lexica

Variable Lexica and the need to be adaptable

Idioms from language, dialect, and jargon As seen

Literary settings

- Some narratives assume background knowledge from a specific period / location
- Tales, Fables, Fantasy, SF... Routinely use an expanded lexicon
- Sometimes types are overloaded (Animals as Agents)



Example with implicit introduction (Lewis Caroll)

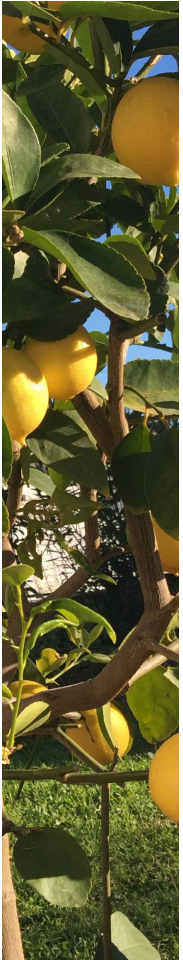
"Beware the Jabberwock, my son !
The jaws that bite, the claws that catch !
Beware the Jubjub bird, and shun
The frumious Bandersnatch !"

Example with explicit introduction (George Martin)

Their driver awaited them beside his *hathay*.
In Westeros, it might have been called an oxcart, though it [...] lacked an ox.
The *hathay* was pulled by a dwarf elephant[...]

Example with in-character introduction (J. K. Rowling)

'I'd like ter see a great Muggle like you stop him,' he said.
'A what ?' said Harry, interested.
'A Muggle,' said Hagrid.
'It's what we call non-magic folk like them.'



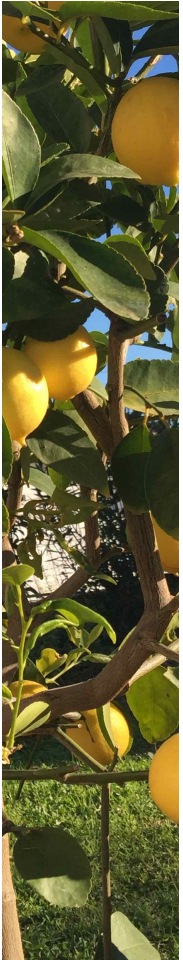
D.26. The Existing Implementation

The syntax and semantics analyser GRAIL

- — Richard Moot's established syntax and semantics parser
- — Wide-coverage, based on categorial grammars
- — Acquires lexicon and rules statistically
- — PROLOG-base, active development
- — Good results on spoken French

Logical programming and additional types

- — No issues when changing the type system
- — Second-order modifiers as a module
- — Actual implementation in λ-DRT rather than λ-calculus

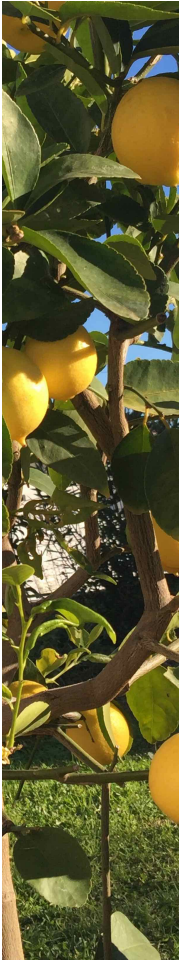


Partial, functioning implementations

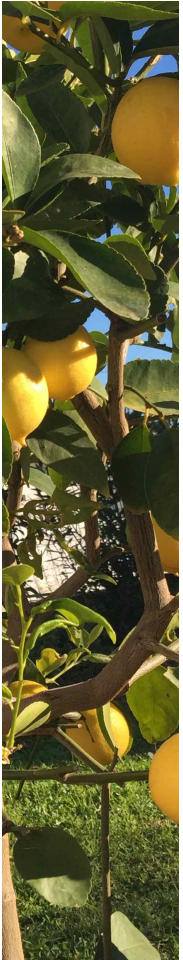
— A proof of concept in 2010 for modifiers
— More recently : application to a specific problem
— Complete, working implementation for a domain
— Travel stories in the Pyrenean region, late XIX^{th}
— Sorts distinguish between places, times, itineraries. . .

What is missing

— Given a type system, Grail can infer the lexicon
— We need a comprehensive type system



E Conclusion

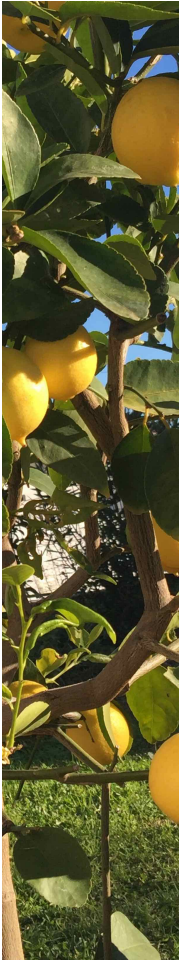


E.1. Modélisation de phénomènes linguistiques spécifiques

— types de base (sortes) pour les restrictions de sélection

— déverbaux :
allumer/allumage
témoigner/témoignage
maquiller/maquillage
garer/garage

— voyageur virtuel :
le sentier descend pendant 20 minutes



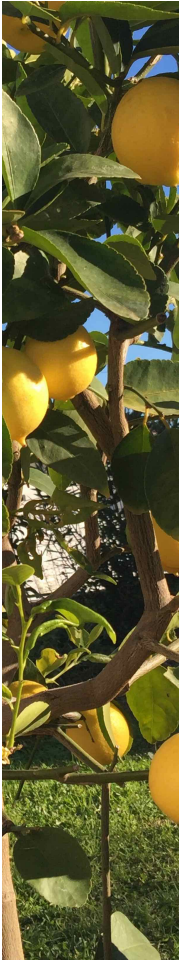
E.2. Questions frontalières entre logique et linguistique

— sémantique des pluriels

— sémantique des noms massifs

— sémantique des événements

— lexiques et ontologies

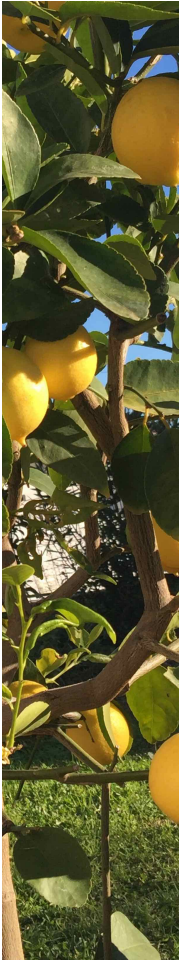


E.3. Questions logiques

— sous typage

— lien théorie des types / calcul des prédicats

— quantification (fonction de choix, epsilon de Hilbert)



E.4. Applications particulières

Expression de la logique dans la langue.

Raisonnement en langage naturel.

Analyse automatique de la structure logique

- d'une argumentation
- d'un débat.