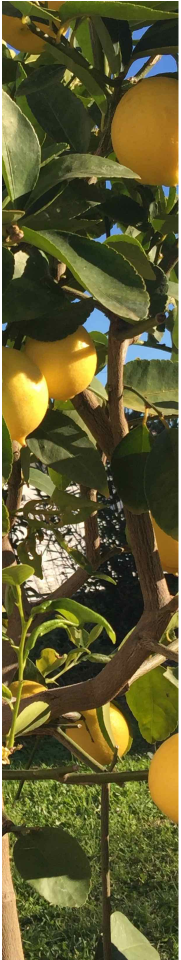
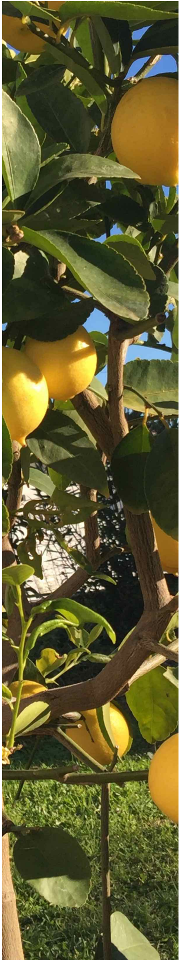


Preuves formelles
et
sémantique du langage naturel

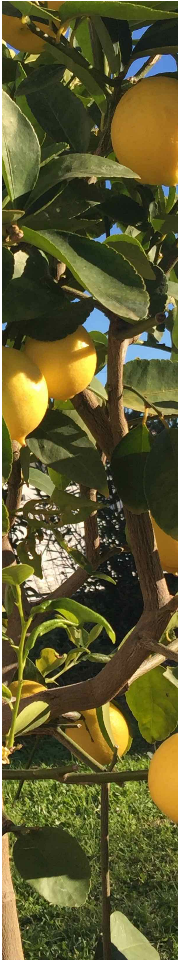
Christian Retoré

Équipe Texte, LIRMM, Université de Montpellier

GDR IM session IM & IA – 12 mars 2019



A Traitement Automatique du Langage Naturel, IA et Informatique Mathématique



A.1. Intelligence Artificielle

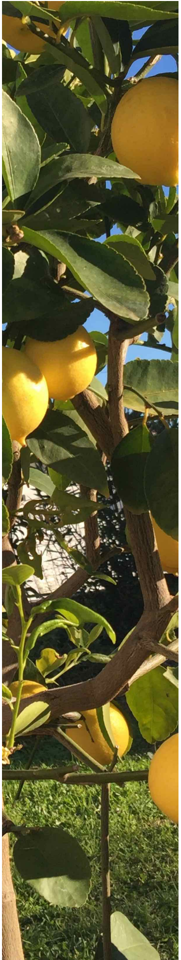
Simuler un comportement humain :

produire et comprendre des énoncés en langage naturel

raisonner (logique)

Test de Turing (1950)

Dans un dialogue écrit avec un ordinateur, 2/3 des individus pensent dialoguer avec un être humain.



A.2. Famille de problèmes à traiter

Deux tâches classiques

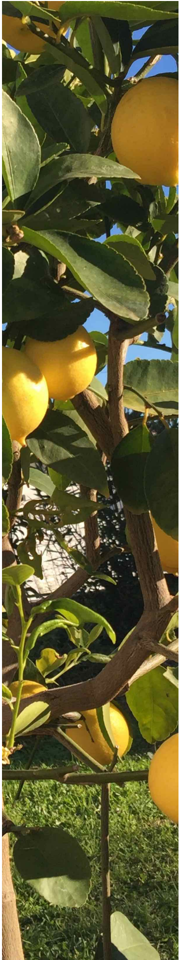
A analyse automatique (de texte, de parole, annoté ou non) résultat ? classification ? représentation thématique ? extraction de relations sémantiques ? **ici : formules logiques**

B génération automatique (de texte, de parole) **ignorée ici** à partir de quoi ?

Domaine ancien, par ex. traduction automatique : depuis 2nde guerre mondiale !

Domaine en rapport :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.3. Applications

Traduction automatique, aide à la traduction. Double 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 ?

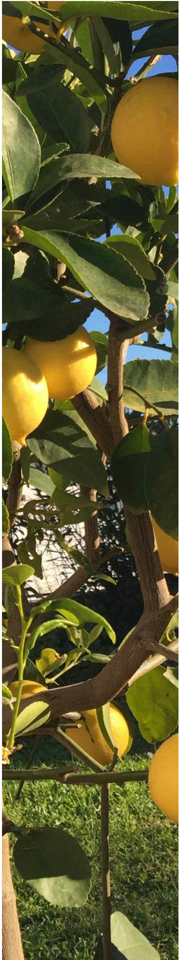
Dialogue homme machine

(2) Les enfants prendront une pizza.

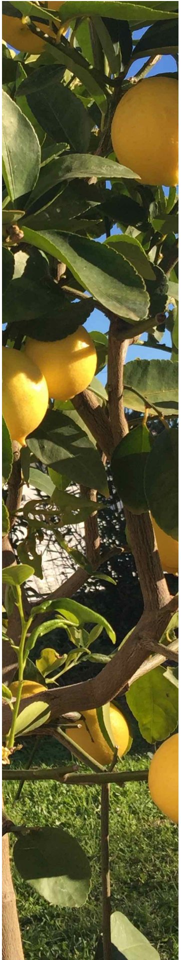
Résumé automatique seule la méthode bête marche
Réponse à des questions (question answering cf. text entailment)

> Geach était il l'élève de Wittgenstein ?

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



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



B.1. Les sons : phonétique/phonologie

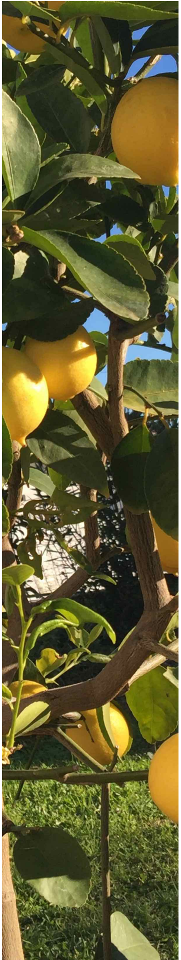
Phonétique Acoustique / système phonatoire/auditif

Techniques : Traitement du signal / médecine

Phonologie Les sons abstraits : système discret

(3) Bali / Paris indistincts pour un japonais

Technique : automates / transducteurs

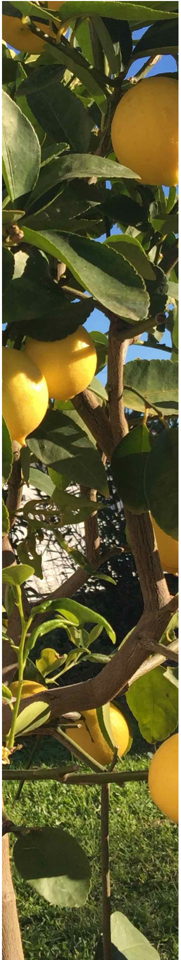


B.2. Morphologie flexionnelle

pas de changement de catégorie

cheval → chevaux
aller → allons
aller → irons

Technique : automates / transducteurs



B.3. Les mots : morphologie dérivationnelle

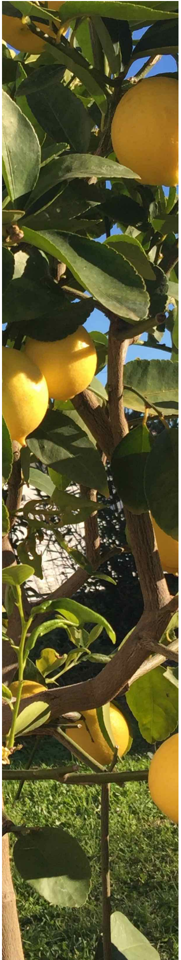
changement de catégorie possible

maison/maisonnette
camion/camionnette
carpe/carpette

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

Technique pour le calcul et l'analyse de la forme : automates / transducteurs

sens : ? ? ?

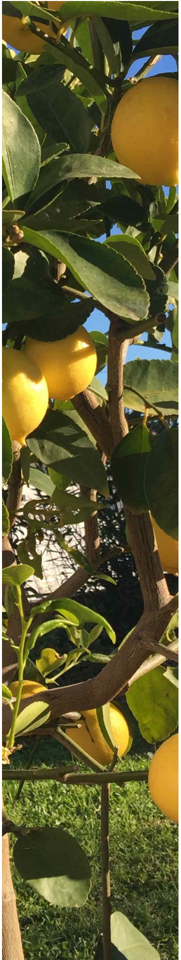


B.4. Étiquetage grammatical

Deviner la catégorie grammaticale (nom, verbe, déterminant, pronom etc.) d'un mot en contexte (Part of Speech tagging) :

La$^{\text{dét}}$ voiture est en panne.
Je la^{pronom} fait réparer.

Automates (règles à la Brill)
ou probas (modèles de Markov cachés)



B.5. Analyse de la phrase (arbre) : syntaxe

** Réparer fais les la le.

*Je fais la réparer

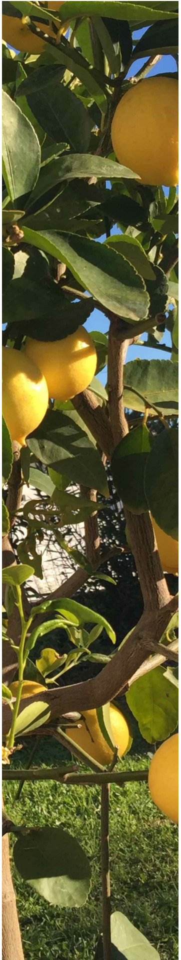
Je la fais réparer

* [[Pierre [mange une]] pomme]

Pierre [mange [une pomme]]

Techniques :

- Grammaires formelles (CFG, TAG etc)
- Déductions formelles dans une logique théorie des types sous structurelle **(cf. ci-après)**
- Arbres + contraintes (dominance, ordre des mots par ex. "le sujet précède le verbe", "le verbe domine ses compléments")



B.6. Le sens des mots : sémantique lexicale

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

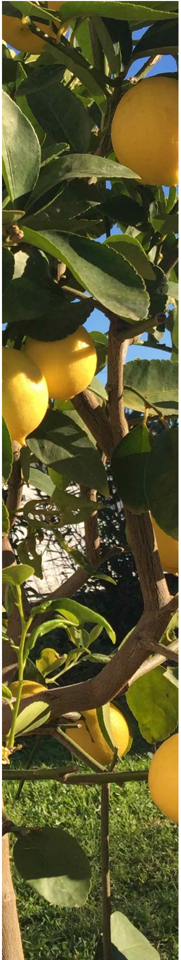
> Rôle télique : être lu, informer, cultiver,
>
> Rôle constitutif : pages, couvertures
>
> Rôle agentif : imprimeur, auteur,

Intégration à la sémantique formelle avec un lambda calcul richement typé **(cf. ci-après)**.

Techniques actuelles : IA, machine learning : vecteurs de mots / word embeddings et apprentissage statistique

sens d'un mot d'un texte = vecteur de mots sur une base (par ex. thésaurus) en fonctions des co-occurrences.

Autre approche : réseau lexical graphe avec 5 10^5 sommets (expressions) et 80 10^6 relations sémantiques de 100 sortes construites par des jeux sérieux JeuxDeMots (Lafourcade)



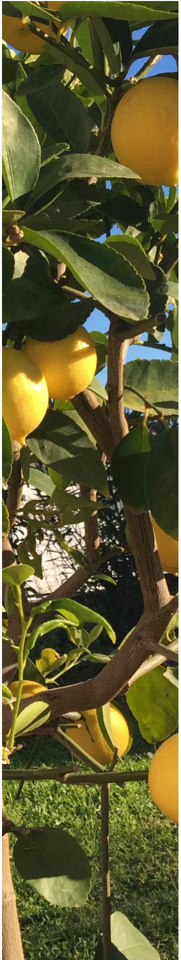
B.7. Le sens d’une phrase : sémantique compositionnelle, formelle, logique

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

Calcul compositionnel (λ calcul) des formules représentant le sens. **(cf. ci-après)**

Interprétation, modèles : mondes possibles dans lequel la phrase est vraie

Problème de de type d’approche : intégration du sens lexical relations entre les sens des mots. **(cf. ci-après)**

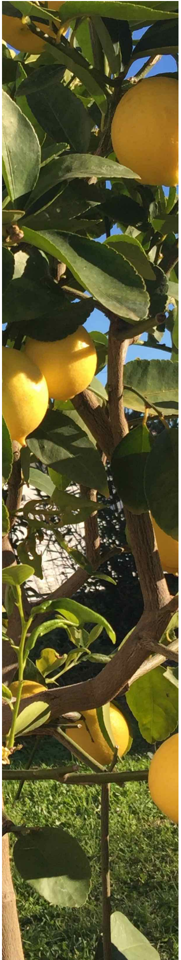


B.8. L’interprétation en en contexte (discours, dialogue) : pragmatique

Prise en compte du contexte linguistique (difficile) et extra linguistique (quasi impossible)

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

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



B.9. Deux notions de sémantique

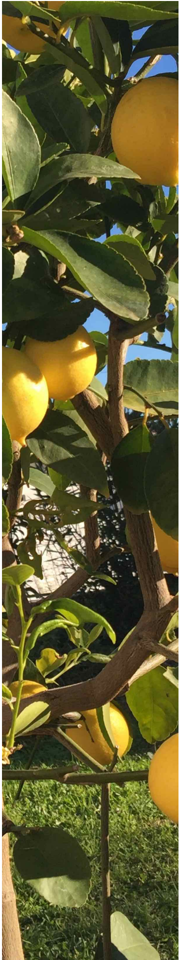
De quoi ça parle (analyse de texte, par des méthodes statistiques, typiquement des vecteurs de mots).

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.



B.10. Approche logique avec des preuves formelles

Dans cet exposé :

syntaxe (déductive)

et sémantique (logique) de la phrase et des mots.

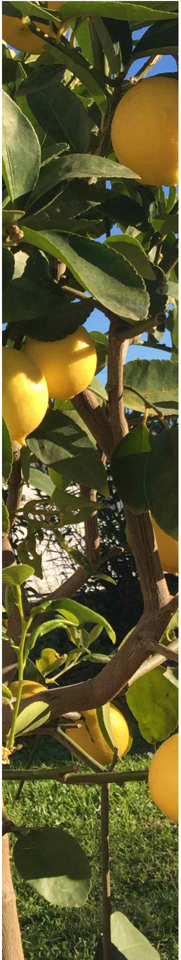
Approche atypique (mais avec théorie des types ;-) :

Nancy, Toulouse,...

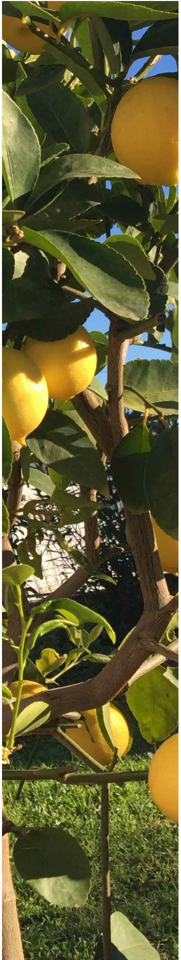
Göteborg

Ohio State University

Tokyo



C Analyse syntaxique orientée vers la structure logique



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

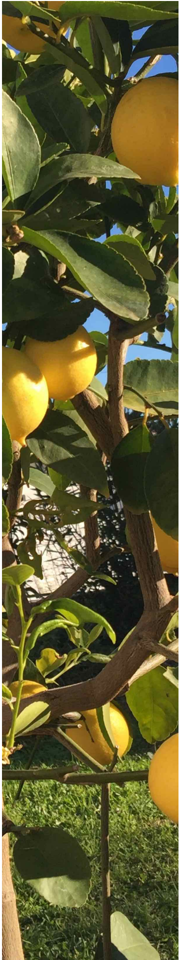
au moins deux hypothèses

A hypothèse la plus à gauche

$$\dfrac{\begin{array}{c} \ldots [A] \ldots \ldots \\ \vdots \\ B \end{array}}{A\backslash B}\ \backslash_i$$

— l'hyp. A est annulée

$$\dfrac{\begin{array}{c} \Delta \\ \vdots \\ A \end{array} \quad \begin{array}{c} \Gamma \\ \vdots \\ A\backslash B \end{array}}{B}\ \backslash_e$$



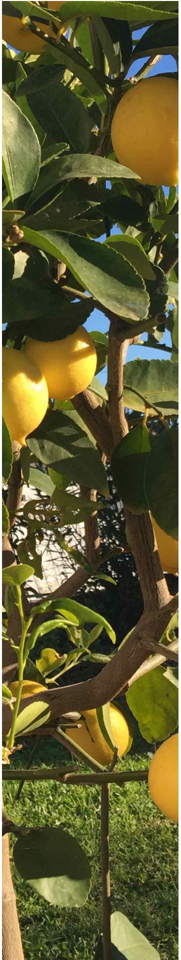
C.2. $A\backslash B$ même chose, mais A est à gauche

au moins deux hyp. libres

A hyp. libre la plus à droite

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

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ B/A\end{array} \quad \begin{array}{c}\Delta \\ \vdots \\ A\end{array}}{B}\,/_e$$



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

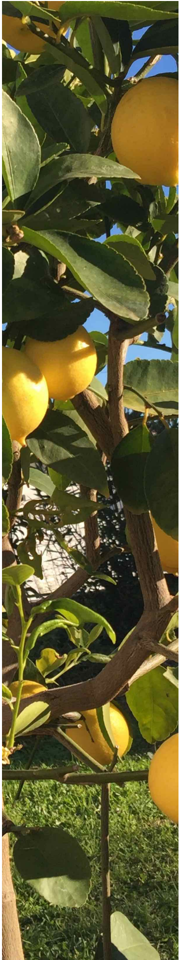
Une phrase est correcte si on peut assigner à chaque mot une catégorie de sorte que la suite des catégories dérive S.

$m_1 \ldots m_n$ est une phrase ssi :

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

Une dérivation est une preuve dans le calcul de Lambek (logique linéaire multiplicative non commutative) : parsing as deduction.

Dans la vraie vie : extension multimodale du calcul de Lambek non asociatif. Les modalités gèrent l'ordre des mots quand il est compliqué, mais ne changent rien du côté sémantique (modalités sémantiquement transparentes).



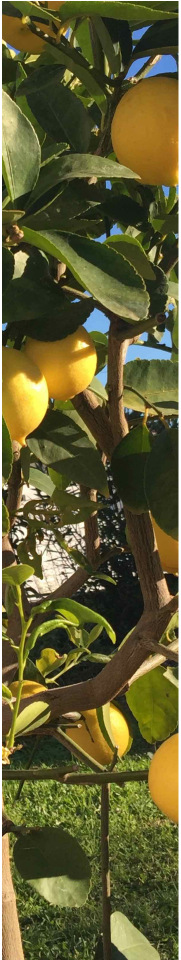
C.4. An example

mot	***catégorie syntaxiqueu***
les	$(S/(np\backslash S))/n$ (subject)
une	$((S/np)\backslash S)/n$ (object)
enfant(s)	n
pizza	n
prendront	$(np\backslash S)/np$

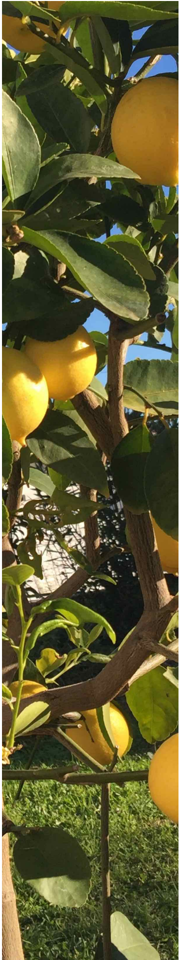
Il y a deux analyses syntaxiques possibles de
Les enfants prendront une pizza.
En voici une :

$\exists\forall$

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



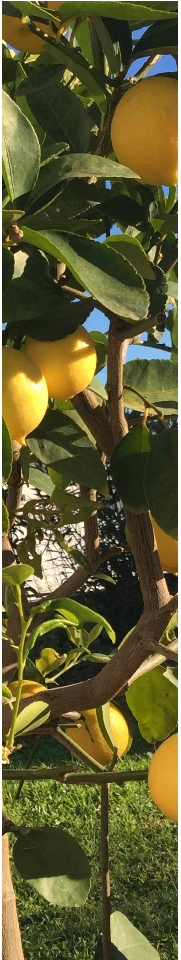
D Analyse sémantique



D.1. Sémantique compositionnelle : principes

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

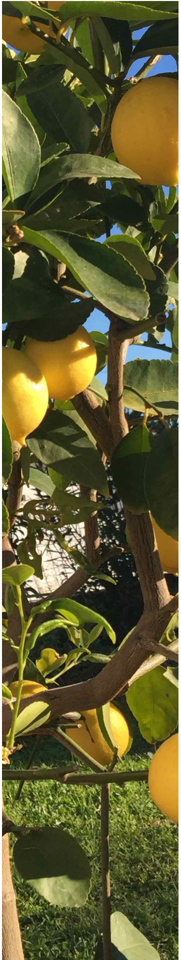
$(\text{Catégorie syntaxique})^*$	$=$	Type sémantique	
S^*	$=$	t	une phrase est une proposition
np^*	$=$	e	un groupe nominal est une entité/individu
n^*	$=$	$e \rightarrow t$	un nom commun est une propriété des entités
$(A \backslash B)^* = (B/A)^*$	$=$	$A \rightarrow B$	propage la traduction



D.2. Des constantes pour les opérations logiques

Il fut des constantes pour les conecteurs et quantificateurs, comme dans le codage à la Church des formules logiques en lambda calcul typé avec deux types (e ou ι entiés ou individus, t ou τ propositions ou valeurs de vérités).

Constante	Type
$\exists$	$(e \rightarrow t) \rightarrow t$
$\forall$	$(e \rightarrow t) \rightarrow t$
$\wedge$	$t \rightarrow (t \rightarrow t)$
$\vee$	$t \rightarrow (t \rightarrow t)$
$\supset$	$t \rightarrow (t \rightarrow t)$

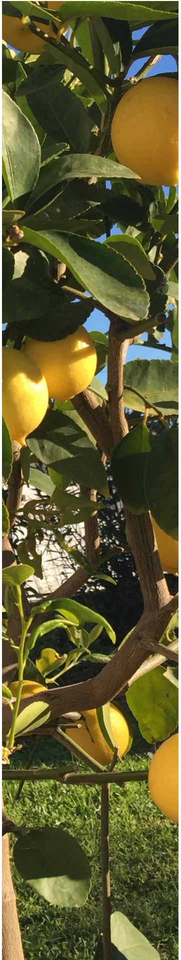


D.3. Des constantes pour les prédicats du langage

La dénotation des mots requiert des prédicats qui sont des constantes du λ calcul.

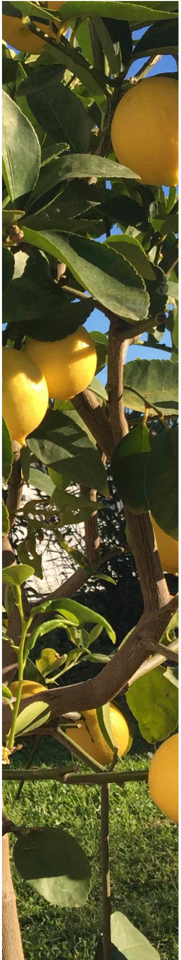
aime	$\lambda x \lambda y\ (\mathrm{aime}\ y)\ x$	$x : e,\ y : e, \mathrm{aime} : e \rightarrow (e \rightarrow t)$
« aime » est un prédicat binaire		

Garance	$\lambda P\ (P\ \mathrm{Garance})$	$P : e \rightarrow t,\ \mathrm{Garance} : e$
« Garance » est décrite comme les propriétés de « Garance »		



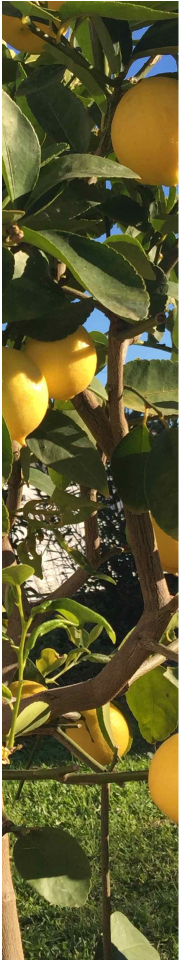
D.4. Sémantique à la Montague : algorithme

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



D.5. 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 $e \to t$ $\lambda x^e(\mathtt{enfant}^{e\to t}\ x)$
pizza	n $e \to t$ $\lambda x^e(\mathtt{pizza}^{e\to t}\ x)$
prendront	$(np\backslash S)/np$ $e \to (e \to t)$ $\lambda y^e\ \lambda x^e\ ((\mathtt{prendront}^{e\to(e\to t)}\ x)y)$

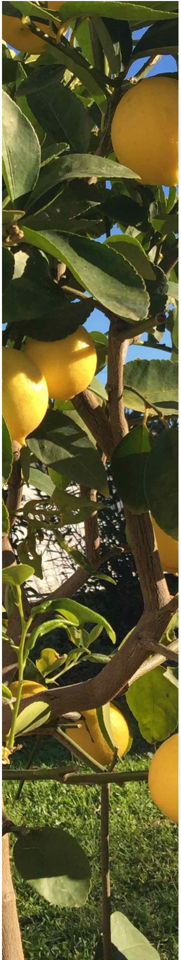


D.6. Analyse syntaxique ∃∀

Il y a deux analyses syntaxiques possibles. Une :

∃∀

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



D.7. Syntaxe $\rightarrow$ λ-terme sémantique de la phrase

$\exists\forall$

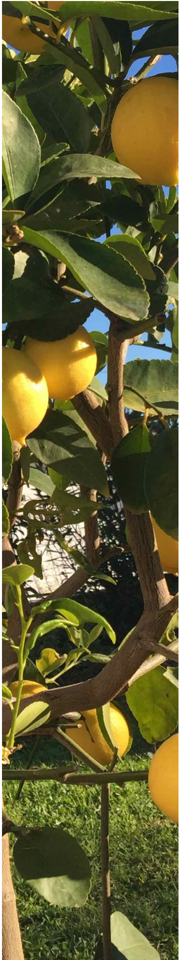
$$
\begin{prooftree}
\AxiomC{$\overset{\textit{les}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}}$}
\AxiomC{$\overset{\textit{enfants}}{(\mathbf{e}\rightarrow\mathbf{t})}$}
\RightLabel{$\rightarrow_e$}
\BinaryInfC{$(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$}
\AxiomC{$\overset{\textit{prendront}}{\mathbf{e}\rightarrow\mathbf{e}\rightarrow\mathbf{t}}$}
\AxiomC{$\overset{o}{[\mathbf{e}]^1}$}
\RightLabel{$\rightarrow_e$}
\BinaryInfC{$\mathbf{e}\rightarrow\mathbf{t}$}
\RightLabel{$\rightarrow_e$}
\BinaryInfC{$\mathbf{t}$}
\RightLabel{$\rightarrow_i(1)$}
\UnaryInfC{$\mathbf{e}\rightarrow\mathbf{t}$}
\AxiomC{$\overset{\textit{une}}{(\mathbf{e}\rightarrow\mathbf{t})\rightarrow(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}}$}
\AxiomC{$\overset{\textit{pizza}}{(\mathbf{e}\rightarrow\mathbf{t})}$}
\RightLabel{$\rightarrow_e$}
\BinaryInfC{$(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$}
\RightLabel{$\rightarrow_e$}
\BinaryInfC{$\mathbf{t}$}
\end{prooftree}
$$

Le λ-terme correspondant est :

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

Il faut encore :

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

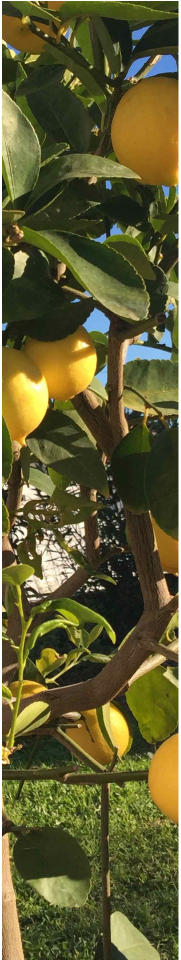


D.8. Calculs, par étapes 1/2

$(\textit{une pizza})$
$= (\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^e(\wedge^{t\to(t\to t)}(P\ x)(Q\ x))))) (\lambda z^e(\mathtt{pizza}^{e\to t}\ z))$
$= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^e(\wedge^{t\to(t\to t)}((\lambda z^e(\mathtt{pizza}^{e\to t}\ z))\ x)(Q\ x)))))$
$= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^e(\wedge^{t\to(t\to t)}((\mathtt{pizza}^{e\to t}\ x)))(Q\ x))))$

$(\textit{les enfants})$
$= (\lambda P^{e\to t}\ \lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda x^e(\Rightarrow^{t\to(t\to t)}\ (P\ x)(Q\ x))))) (\lambda u^e(\mathtt{enfant}^{e\to t}\ u))$
$= (\lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda x^e(\Rightarrow^{t\to(t\to t)}\ ((\lambda u^e(\mathtt{enfant}^{e\to t}\ u))\ x)(Q\ x)))))$
$= (\lambda Q^{e\to t}\ (\forall^{(e\to t)\to t}\ (\lambda x^e(\Rightarrow^{t\to(t\to t)}\ (\mathtt{enfant}^{e\to t}\ x)(Q\ x)))))$

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

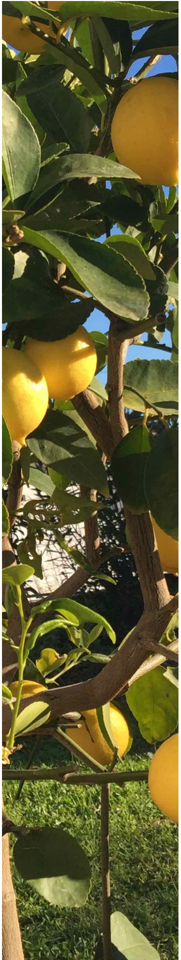


D.9. Calculs, par étapes 2/2

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

ce qui s'écrit communément :

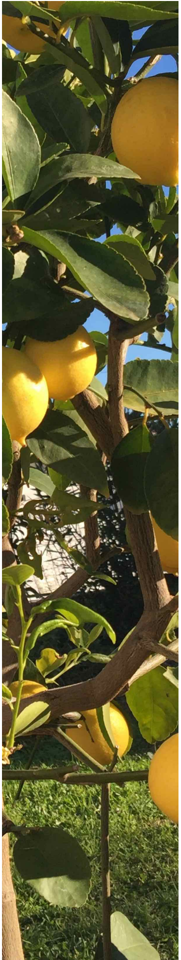
$$\exists x.\ \mathit{pizza}(x) \wedge \forall w.\ (\mathit{enfant}(w) \Rightarrow \mathit{prendront}(w,x))$$



D.10. Avec l’autre analyse syntaxique...

$\forall\exists$

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



D.11. Une autre structure sémantique....

Qui correspond à l'analyse : $\forall\exists$

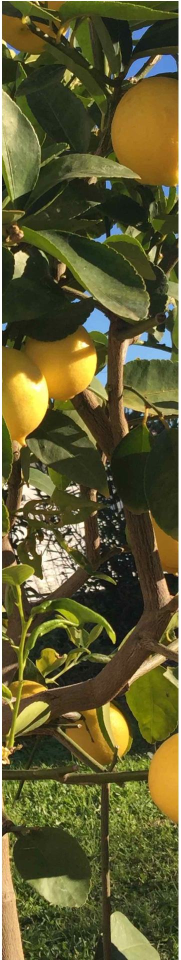
$$
\begin{prooftree}
\AxiomC{$\overset{\textit{les}}{(\mathbf{e}\to\mathbf{t})\to(\mathbf{e}\to\mathbf{t})\to\mathbf{t}}$}
\AxiomC{$\overset{\textit{enfants}}{(\mathbf{e}\to\mathbf{t})}$}
\RightLabel{$\to_e$}
\BinaryInfC{$(\mathbf{e}\to\mathbf{t})\to\mathbf{t}$}
\AxiomC{$\overset{s}{[\ \mathbf{e}]^1}$}
\AxiomC{$\overset{\textit{prendront}}{\mathbf{e}\to\mathbf{e}\to\mathbf{t}}$}
\AxiomC{$\overset{o}{[\mathbf{e}]^2}$}
\RightLabel{$\to_e$}
\BinaryInfC{$\mathbf{e}\to\mathbf{t}$}
\RightLabel{$\to_e$}
\BinaryInfC{$\mathbf{t}$}
\RightLabel{$\to_i(2)$}
\UnaryInfC{$\mathbf{e}\to\mathbf{t}$}
\AxiomC{$\overset{\textit{une}}{(\mathbf{e}\to\mathbf{t})\to(\mathbf{e}\to\mathbf{t})\to\mathbf{t}}$}
\AxiomC{$\overset{\textit{pizza}}{(\mathbf{e}\to\mathbf{t})}$}
\RightLabel{$\to_e$}
\BinaryInfC{$(\mathbf{e}\to\mathbf{t})\to\mathbf{t}$}
\RightLabel{$\to_e$}
\BinaryInfC{$\mathbf{t}$}
\RightLabel{$\to_i(1)$}
\UnaryInfC{$\mathbf{e}\to\mathbf{t}$}
\RightLabel{$\to_e$}
\BinaryInfC{$\mathbf{t}$}
\end{prooftree}
$$

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

($(\textit{une pizza})$ et $(\textit{les enfants})$ déjà faits)



D.12. Calculs (bis repetita placent)

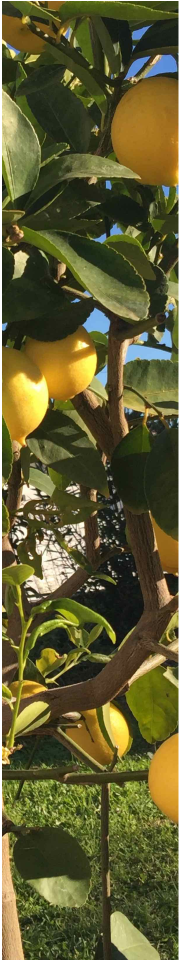
$$(\mathit{une\ pizza})(\lambda o\ ((\mathit{prendront}\ o)\ s))$$
$$= (\lambda Q^{e\to t}\ (\exists^{(e\to t)\to t}\ (\lambda x^{e}(\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))(Q\ x))))$$
$$(\lambda o\ (((\lambda y^{e}\ \lambda x^{e}\ ((\texttt{prendront}^{e\to(e\to t)}\ x)\ y))\ o)\ s)))$$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

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

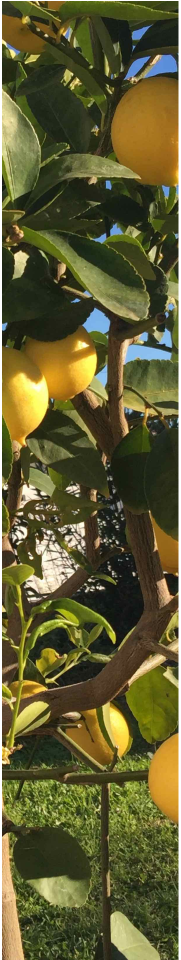
$$= (\forall^{(e\to t)\to t}\ (\lambda u^{e}(\Rightarrow^{t\to(t\to t)}\ (\texttt{enfants}^{e\to t}\ u)$$
$$(\exists^{(e\to t)\to t}\ (\lambda x^{e}.\ (\wedge^{t\to(t\to t)}((\texttt{pizza}^{e\to t}\ x)))((\texttt{prendront}^{e\to(e\to t)}\ u)\ x))))))$$

ce qui s'écrit communément :

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



E Prise en compte du sens lexical



E.1. Restriction de sélection

Un prédicat ne se compose qu'avec des arguments appropriés :

(6) # La chaise a aboyé.

(7) ? ? ? La chaise n'a pas aboyé.

(8) Le chien des voisins a aboyé.

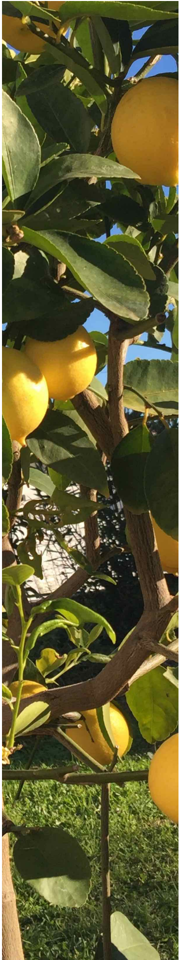
(9) ? Un renard a aboyé.

Entités de Montague $\rightarrow$ types de base.

Conflit de types :

$\mathrm{aboyer}^{\mathrm{canidé}}(\mathit{Rex}^{\mathrm{canidé}})$ OK

$\mathrm{aboyer}^{\mathrm{canidé}}(\mathit{la\_chaise}^{\mathrm{mobilier}})$ pas OK



E.2. Coercitions

Sorte d'adaptation du sens au contexte :

(10) Le sergent a aboyé après la nouvelle recrue.

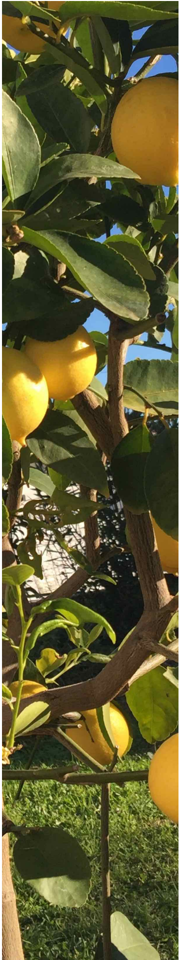
(11) J'ai perdu mon livre. (matériel)

(12) J'avais beaucoup aimé ce livre. (contenu)

(13) Liverpool est un port. (lieu)

(14) Liverpool est cosmopolite. (gens)

(15) Liverpool a battu Arsenal. (club)



E.3. Co-prédication

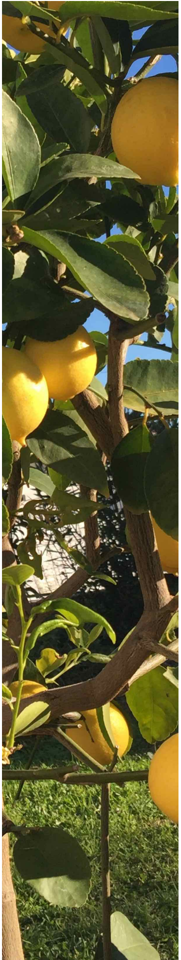
Affirmer des choses de plusieurs aspects :

(16) J’ai perdu ce livre que j’avais beaucoup aimé.

(17) Liverpool est un port et vote labour.

(18) * Liverpool est un port et a battu Arsenal.

Le sens "club de foot" est incompatible avec les autres.



E.4. Les lambda termes sémantiques (système F)

Types :

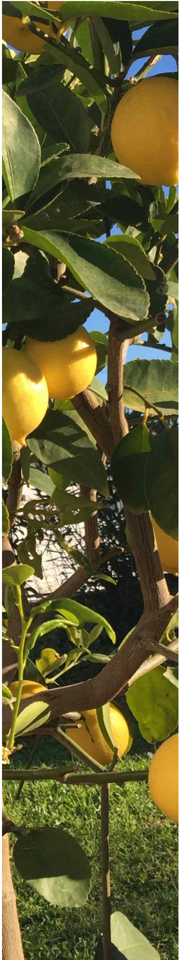
- des types de base e_i (sortes)
- des variables de type $\alpha, \beta, \ldots$
- types fonctionnels $a \to b$
- type quantifiés : $\Pi\alpha.\ T[\alpha]$

Termes :

- variables, constantes
- $f^{u \to v} t^u$
- $\lambda x^u.\ t^v : u \to v$
- $t^{\Pi\alpha.T}\{U\} : T[\alpha := U]$

Beta reduction :

$$(\Lambda\alpha.\ t^T)\{U\} \rightsquigarrow t[\alpha := U]$$



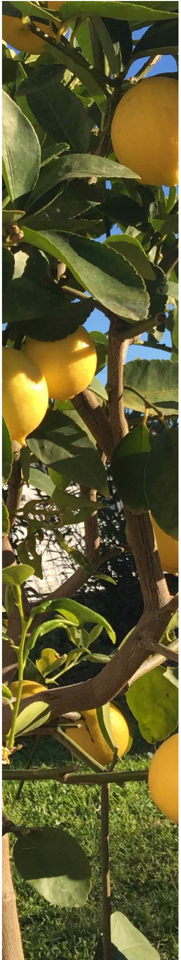
E.5. Structure du lexique

$$(\lambda x^U.\ t^V)a^U \rightsquigarrow t[x^U := a^U]$$

Lexicon : word $\rightarrow$
- terme usuel (principal)
- termes optionnels pour régler les conflits de types
- termes optionnels déclarés rigide ou flexible

terme optionnel rigide :
l'utilisation de ce terme optionnel (de ce sens) est incompatible avec tout autre terme optionnel (avec tout autre sens).

Exemple : pour une ville le sens "club de foot"
est incompatible avec les autres sens.



E.6. Conjunction for co-predication

! Polymorphic conjunction :

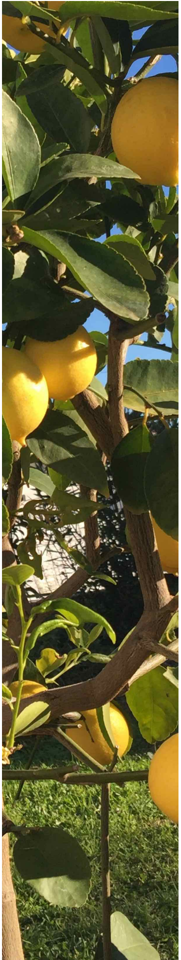
Given predicates $P^{\alpha\to\mathbf{t}}$, $Q^{\beta\to\mathbf{t}}$ over entities of respective types α, β,

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

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



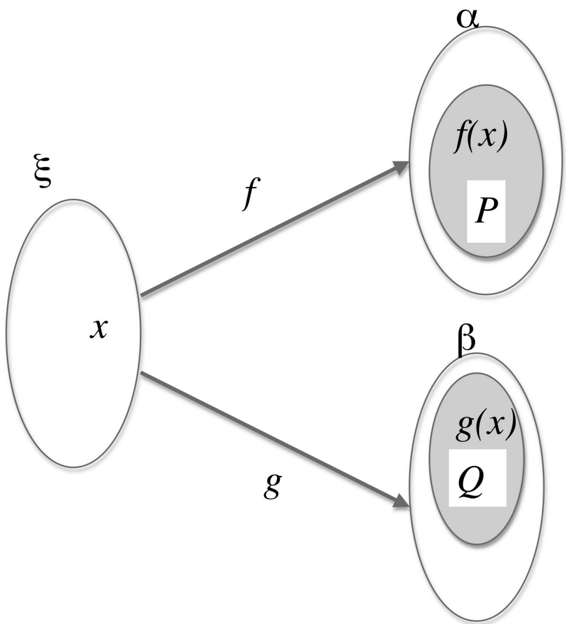
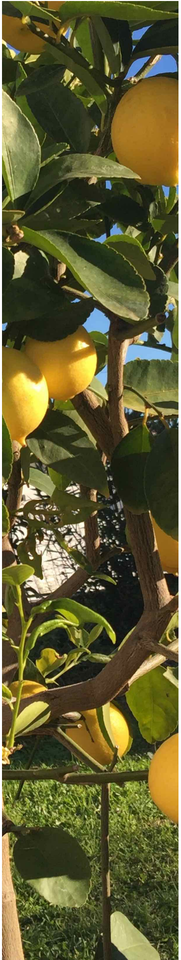


FIGURE 1 – Polymorphic conjunction : $P(f(x))\&Q(g(x))$ with $x : \xi$, $f : \xi \to \alpha$, $g : \xi \to \beta$.

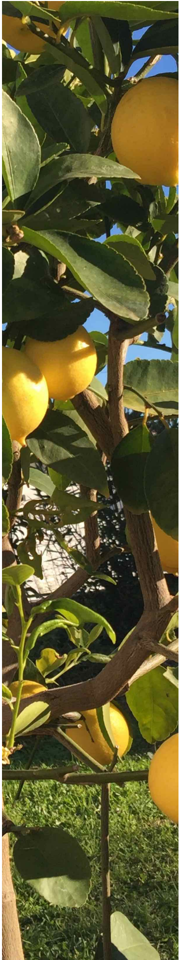


E.7. An example of a lexicon

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) F (football club)

Lexical item	Main λ-term	Modifiers
Liverpool	$Liverpool^T$	$Id_T : T \to T(Fl)$ $t_2 : T \to P(Fl)$ $t_3 : T \to Pl(Fl)$ $t_4 : T \to F(R)$
is_a_big_place	$big\_place : Pl \to \mathbf{t}$	
voted	$voted : P \to \mathbf{t}$	
won	$won : F \to \mathbf{t}$	



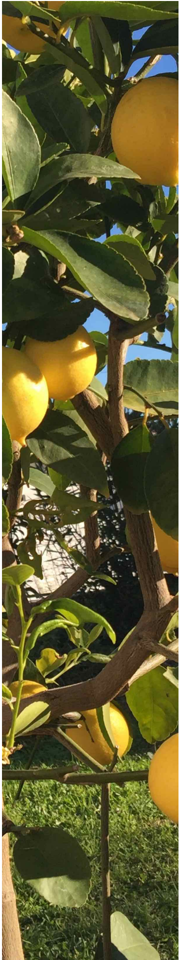
E.8. Examples of coercions

1. *Liverpool is a big place*
2. *Liverpool voted.*
3. *Liverpool won.*

1 Type mismatch in $big\_place^{Pl\rightarrow\mathbf{t}}(Liverpool^{T}))$, resolved using t_3 :

$$big\_place^{Pl\rightarrow\mathbf{t}}(t_3^{T\rightarrow Pl}Liverpool^{T}))$$

2 & 3 The same, can be done using other coercions t_2, t_4 for examples 2 and 3.



E.9. Examples of coercions and copredications

1. *Liverpool is a big place and voted.*
2. *Liverpool is a big place and won.*

1 We use the polymorphic conjunction operator, $\&^{\Pi}$.

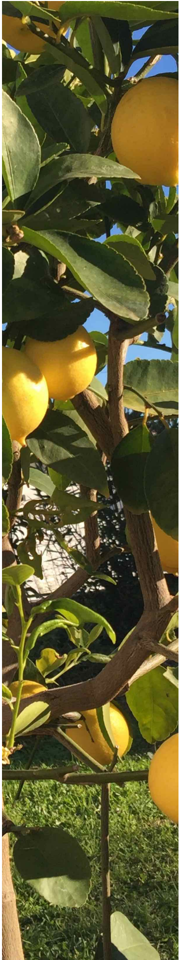
$$\Lambda\xi \lambda x^{\xi}\, \lambda f^{\xi\to\alpha} \lambda g^{\xi\to\beta} (\mathrm{and}^{(\mathbf{t}\to\mathbf{t})\to\mathbf{t}}\, (big\_place\, (f\ x))(voted\, (g\ x)))$$

After application, we have :

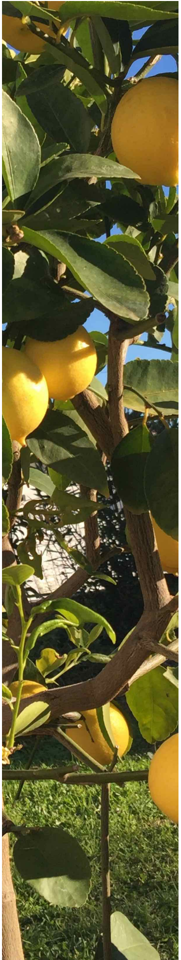
$$(\mathrm{and}\ (big\_place^{Pl\to\mathbf{t}}\ (t_3^{T\to Pl}\ Liverpool^{T}))$$

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

2 we cannot do as above, because using t_4 which is rigid prevents from using t_3.



F Liens avec d’autres domaines : outils de TALN, logique, linguistique



F.1. Outil : plateforme Grail (Moot) – syntaxe

$\sim$ Boxer (Bos) mais Grail Français et MMCG plutôt que CCG

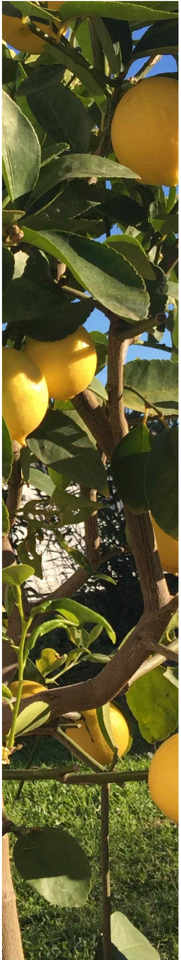
Idées mises en oeuvre dans la plateforme Grail (Moot)

Grammaires acquise sur corpus annoté (corpus P7 Le Monde A. Abeillé) récemment par Deep Learning

Autres possibilité clusters de mots similaires apprentissage par typage exact à la Gold / Kanazawa

Forme d’inférence grammaticale avec théorème de convergence. Beau sujet. Projet ARC INRIA Gracq 2000-2003 apprentissage de CG avec Isabelle Tellier, prof à Orléans, prématurément disparue.

Supertagging N-best optimal pour N=7 (un des premiers)



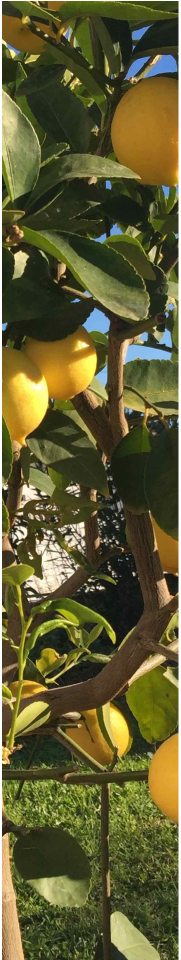
F.2. Outil : plateforme Grail (Moot) – sémantique

$\sim$ Boxer (Bos) mais Grail lambda calcul typé plutôt que calcul *ad hoc* sur la structure syntaxique

Grail produit des Discourse Representation Structures $\sim$ équivalent à des formules logiques, mais permet de gérer les pronoms et leurs référents.

Lambda termes sémantiques :

— saisis manuellement pour les mots grammaticaux
— sinon
$chat : \lambda x^e.\ chat(x)$
$regarde : \lambda x^e . \lambda y^e\ ((regarde y) x)$
— machine learning ?



F.3. Outil : plateforme Grail (Moot) – sémantique lexicale

Pas d’analogue dans Boxer.

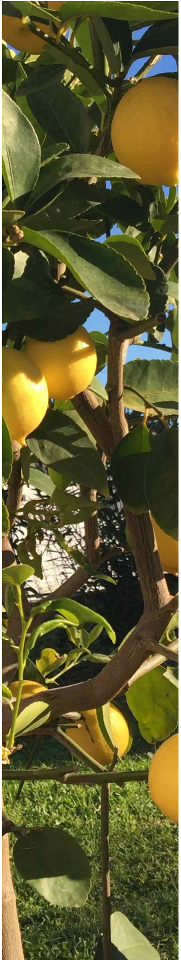
Coercitions pour MGL :

- tests saisis manuellement (e.g verbes de mouvements)
- acquisition :
  - réseaux lexical JeuxDeMots
  - Machine Learning ? ? ?

Créneau pour Grail : quand une analyse profonde et logique est nécessaire, ou pour des linguistes cf. ci-après.

Exemples : dialogue homme machine, analyse d’un raisonnement, d’une argumentation d’un débat en langue naturel.

A savoir : argument mining qui classe les interventions en pour/contre ignore les interventions contenant des négations ! ! !



F.4. Calculabilité et complexité

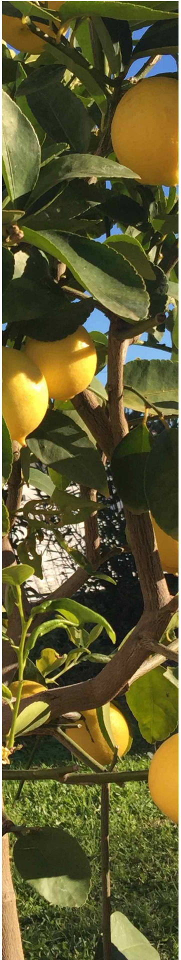
Choix des catégories syntaxiques
$\rightarrow$ Supertagging (linéaire)

Analyse syntaxique des 7 meilleurs :
analyse syntaxique polynomiale (pour certains systèmes)
si le degré d'imbrication est borné (ce qui est toujours le cas pour un lexique syntaxique, qui est grand mais fini) (Pentus 2010) — mais prouvabilité NP complète.

Choix des coercions en cas de conflits de types :
exponentiel (mais 2 ou 3 choix dans une phrase).

Calcul de la formule logique par β-réduction : polynomial car dans Soft Linear Logic (Lafont) : pas d'exponentielles.

En pratique il n'y a pas de problème de complexité hormis dans le choix des catégories syntaxiques (problème résolu par super tagging).



F.5. Interrogation sur la complexité

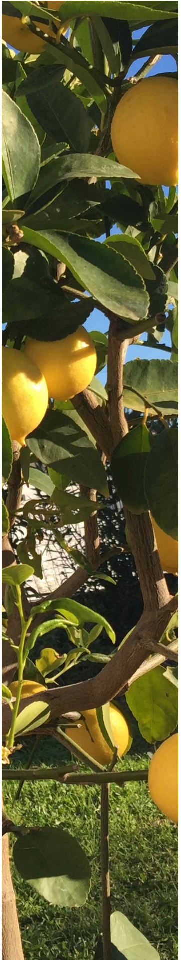
Phrase 10 mots (à l'oral) et 25 mots (à l'écrit) sur 300.000 symboles ($\neq$ bioinfo ! "phrases" de millions de "mots" parmi quatre)

Analyse asympotique dans le pire des cas inadaptée.

Quelle serait une bonne description de la complexité de l'analyse ?

Complexité en moyenne (quelle distribution) ?

Taille de la grammaire ou du lexique ?



F.6. Liens avec la linguistique formelle

Formaliser, calculer, prédire $\rightarrow$ verification / modification des modèles linguistiques par exmple pour des raisons de préférences complexité.

Syntaxe : formalisation du cadre Chomskyien en insistant sur l'interface avec la sémantique.

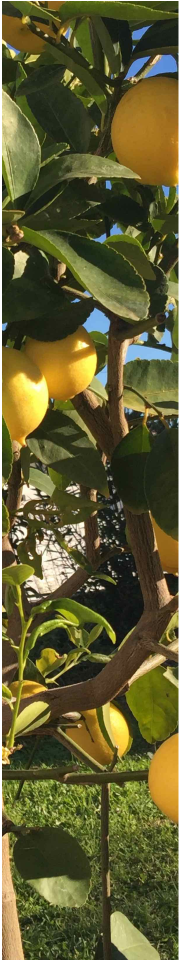
Sémantique formelle :

- pluriels
- quantification *tout/chaque*
- valeur argumentative (Anscombre et Ducrot)

  Tu peux me prêter de largent ?

  J'ai **peu** / **un peu** d'argent.

  non / **oui**



F.7. De jolies questions de logique mathématique

Expression des formules en théorie des types à la Church : expression des groupes (pluriels), des quantités mesurables,...

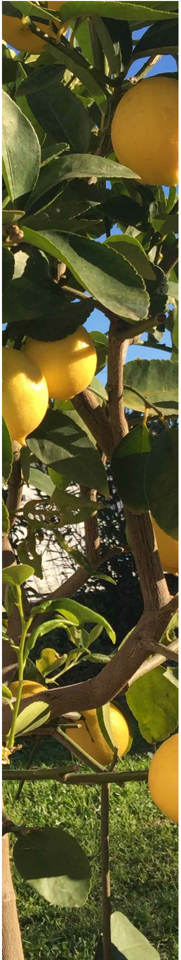
Liens entre calcul des prédicats d'ordre supérieur (multisorte) et théorie des types : "$a : A$" non réfutable // "$\tilde{A}(a)$" réfutable : "$\neg\tilde{A}(a)$".

Quantification avec les opérateurs de Hilbert ε et τ. proche de la structure syntaxique de la phrase (over-binding, ce quantificateur, même "enfoui" dans la structure syntaxique peut porter sur toute la formule). Logique bien plus expressive que les logiques d'ordre supérieur, ou qu la théorie des type : à comparer avec les quantificateurs branchants de Henkin, et les quantificateurs de portée sous spécifiée.

Logique modale S4 avec premier ordre :

$$\underline{tout}\, x.\ P(x) = \Box\Diamond\Box\forall x.\ P(x)$$

Les mondes exceptionnels où $\forall x.\ P(x)$ est faux sont rares — au sens topologique sur $\{0,1\}^*$ avec comme ouverts de base les ensembles clos vers le haut.

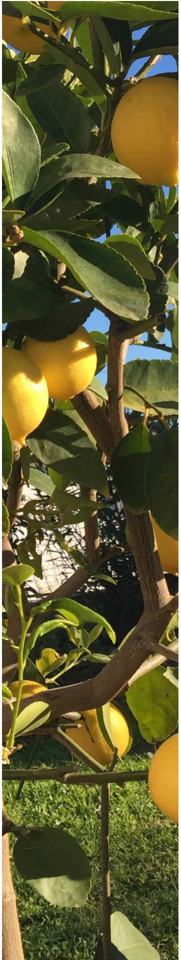


F.8. Projet : sémantique argumentative

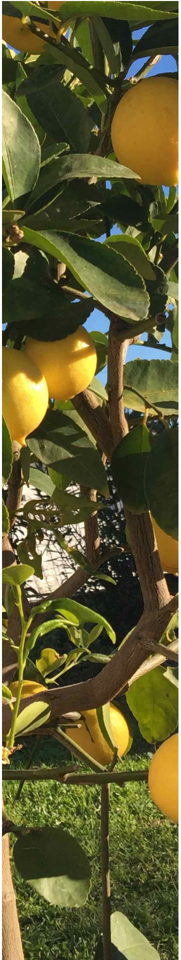
Sémantique des phrases en termes de dialogue argumentatifs : justifications / réfutations, à rapprocher de :

- "proof theoretical semantics" (Dummet, Prawitz, Francez) mais avec une interprétation des énoncés non démontrables
- "dialogical logic" (Lorenz, Lorenzen, Felscher)
- "natural logic" (Moss, Muskens)).

Application à l'analyse automatique de raisonnement, d'argumentation, de débats.



G Conclusion



Une approche logique atypique
mais fortement reliée à d'autres domaines.

Concernant le traitement automatique des langues

- techniques d'apprentissage standard bienvenues pour construire les lexiques nécessaires
- des outils d'analyse automatique du langage naturel pour des utilisations particulières

Des connexions prometteuses avec d'autres domaines

- linguistique formelle (pluriels, noms massifs, quantification, négation, valeur argumentative)
- logique mathématique (théorie des types, logique d'ordre supérieur, quantification, modalités)

Projet

- Expression de la logique dans la langue
- Analyse automatique de la structure logique d'une argumentation ou d'un débat