



Notes on Grammar Extraction and the French Treebank

Richard Moot (SIGNES)

Introduction

- ◆ Last time, I discussed some of the basic aspects of the lexicon for categorial grammars which has been extracted from the Paris VII treebank.
- ◆ The session I will talk about some other things (control verbs, passives, adjectives and adverbs) and link them to the semantics.
- ◆ I will start with some brief revision of categorial grammar and semantics.

Types

Inductive Definition

- ◆ Basic types e (for entity) and t (for truth value)
- ◆ If α and β are types, then $\alpha \rightarrow \beta$ is a type

Terms

- ◆ For each type α , there is a (countably infinite) number of variables x, y, z, \dots which are terms of type α .
- ◆ For each type α , there is a (countably infinite) number of constants a, b, c, \dots which are terms of type α .
- ◆ If x is term of type $\alpha \rightarrow \beta$ and y is a term of type α then $(x y)$ is a term of type β .
- ◆ If y is a term of type β and x is a variable of type α , then $\lambda x.y$ is a term of type $\alpha \rightarrow \beta$

Terms

◆ Some useful constants

$\wedge, \vee, \rightarrow$	$t \rightarrow (t \rightarrow t)$
\forall, \exists	$(e \rightarrow t) \rightarrow t$
\perp	$(e \rightarrow t) \rightarrow e$

Notational Conventions

- ◆ These notational conventions are useful for people familiar with predicate logic and Montague grammar.
- ◆ We will write $(x \wedge y)$ instead of $((\wedge y) x)$ and adopt a similar convention for the other boolean connectives
- ◆ We will write $\iota x.t$ instead of $\iota(\lambda x.t)$ and adopt a similar convention for \forall and \exists
- ◆ We will write $p(x,y)$ instead of $((p y) x)$ for constants p (in general $p(x_1, \dots, x_n)$ for $(\dots(p x_n) \dots x_1)$)

Notational Conventions

- ◆ $\lambda z.\lambda y.\lambda x.(((f\ z)\ y)\ x) \equiv \lambda z.\lambda y.\lambda x.f(x,y,z)$
- ◆ $\forall(\lambda x.\exists(\lambda y.((\text{love } y)\ x))) \equiv \forall x.\exists y.\text{love}(x,y)$
- ◆ $\forall(\lambda x.(\rightarrow(\text{man } x))\ \text{sleep } x)) \equiv \forall x.\text{man}(x) \rightarrow \text{sleep}(x)$

Formulas as types

- ◆ This is the standard way of assigning types to formulas in categorial grammar

In many of the examples, I will assign e to both np and vp , I will leave the changes required to turn these into terms of type $(e \rightarrow t) \rightarrow t$ as an exercise

- ◆ Exception 1: np is lifted from e to $(e \rightarrow t) \rightarrow t$
- ◆ Exception 2: pp is assigned the same type as np

Formula	Type
$\text{type}(np)$	$(e \rightarrow t) \rightarrow t$
$\text{type}(pp)$	$(e \rightarrow t) \rightarrow t$
$\text{type}(n)$	$e \rightarrow t$
$\text{type}(s)$	t
$\text{type}(B/A)$	$\text{type}(A) \rightarrow \text{type}(B)$
$\text{type}(A \setminus B)$	$\text{type}(A) \rightarrow \text{type}(B)$

Formulas as types

- ◆ Exception 1: from e to $(e \rightarrow t) \rightarrow t$
- ◆ Exception 2: the same type as np
- ◆ To simplify the examples which follow, I will use e as the type of np and pp in the examples (exercise: give the correct lambda terms)

In many of the examples, I will assign e to both np and vp , I will leave the changes required to turn these into terms of type $(e \rightarrow t) \rightarrow t$ as an exercise

Formula	Type
$\text{type}(np)$	e
$\text{type}(pp)$	e
$\text{type}(n)$	$e \rightarrow t$
$\text{type}(s)$	t
$\text{type}(B/A)$	$\text{type}(A) \rightarrow \text{type}(B)$
$\text{type}(A \setminus B)$	$\text{type}(A) \rightarrow \text{type}(B)$

Proofs as terms

◆ Proofs in categorial grammar correspond to lambda terms

$$\frac{t:A/B \quad u:B}{(t \ u):A} \quad \frac{u:B \quad t:B \setminus A}{(t \ u):A}$$

◆ These lambda terms abstract away from the directions of the implications.

$$\frac{[x:B] \quad \vdots \quad t:A}{A/B:\lambda x.t} \quad \frac{[x:B] \quad \vdots \quad t:A}{B \setminus A:\lambda x.t}$$

Proofs as terms

- ◆ Type lifting is now a simple consequence of the logical rules.

$$\frac{t:A/B \quad u:B}{(t \ u):A} \quad \frac{u:B \quad t:B \setminus A}{(t \ u):A}$$

$x:np$

$$\frac{[x:B] \quad \vdots \quad t:A}{A/B:\lambda x.t} \quad \frac{[x:B] \quad \vdots \quad t:A}{B \setminus A:\lambda x.t}$$

Proofs as terms

- ◆ Type lifting is now a simple consequence of the logical rules.

$$\frac{t:A/B \quad u:B}{(t \ u):A}$$

$$\frac{u:B \quad t:B \setminus A}{(t \ u):A}$$

$$\frac{x:np \quad P:(np \setminus s)}{(P \ x):s}$$

$$\frac{[x:B] \quad \vdots \quad t:A}{A/B:\lambda x.t}$$

$$\frac{[x:B] \quad \vdots \quad t:A}{B \setminus A:\lambda x.t}$$

Proofs as terms

- Type lifting is now a simple consequence of the logical rules.

$$\frac{t:A/B \quad u:B}{(t \ u):A} \quad \frac{u:B \quad t:B \setminus A}{(t \ u):A}$$

$$\frac{\frac{x:np \quad [P:(np \setminus s)]}{(P \ x):s}}{\lambda P.(P \ x):s / (np \setminus s)}$$

$$\frac{[x:B] \quad \vdots \quad t:A}{A/B:\lambda x.t}$$

$$\frac{[x:B] \quad \vdots \quad t:A}{B \setminus A:\lambda x.t}$$

Verbs

(np \ s)

dorment

$\lambda x.$ dormir(x)

$\lambda xe.$ dormir(e,x)

(np \ s) / np

dénonçait

$\lambda y.\lambda x.$ dénoncer(x,y)

$\lambda yxe.$ dénoncer(e,x,y)

(np \ s) / pp_{de}

dépendent

$\lambda y.\lambda x.$ dépendre_de(x,y)

$\lambda yxe.$ dépendre_de(e,x,y)

Verbs

$(np \setminus s) / (np \setminus S_{ppart})$

est, a

Note that there is nothing which prohibits assigning $pouvoir(x,y)$ as a semantics, however the current semantics correctly predicts that the subject of $pouvoir$ is also the subject of the infinitive

$(np \setminus s) / (np \setminus S_{inf})$

peut (infinitive)

$\lambda yx.pouvoir(x,(yx))$

$(np \setminus s) / (np \setminus S_{deinf})$

cessera (de infinitive)

$(np \setminus s) / (np \setminus S_{ainf})$

tarde (à infinitive)

Verbs

np / n

n

(np \ s) / (np \ S_{inf})

Le

parquet

peut

$\lambda y. \iota x (y \ x)$

$\lambda z. \text{parquet}(z)$

$\lambda yx. \text{pouvoir}(x, (yx))$

(np \ S_{inf}) / np

np / n

n

saisir

la

justice

$\lambda yx. \text{saisir}(x, y)$

$\lambda y. \iota x (y \ x)$

$\lambda z. \text{justice}(z)$

Verbs

np / n

n

(np \ s) / (np \ S_{inf})

Le

parquet

peut

$\lambda y. \iota x (y \ x)$

$\lambda z. \text{parquet}(z)$

$\lambda yx. \text{pouvoir}(x, (yx))$

(np \ S_{inf}) / np

np

saisir

la justice

$\lambda yx. \text{saisir}(x, y)$

$(\lambda y. \iota x (y \ x) \ \lambda z. \text{justice}(z))$

Verbs

np / n

n

(np \ s) / (np \ S_{inf})

Le

parquet

peut

$\lambda y. \iota x (y \ x)$

$\lambda z. \text{parquet}(z)$

$\lambda yx. \text{pouvoir}(x, (yx))$

(np \ S_{inf}) / np

np

saisir

la justice

$\lambda yx. \text{saisir}(x, y)$

$\iota x. (\lambda z. \text{justice}(z) \ x)$

Verbs

np / n

n

(np \ s) / (np \ S_{inf})

Le

parquet

peut

$\lambda y. \iota x (y \ x)$

$\lambda z. \text{parquet}(z)$

$\lambda yx. \text{pouvoir}(x, (yx))$

(np \ S_{inf}) / np

np

saisir

la justice

$\lambda yx. \text{saisir}(x, y)$

$\iota x. \text{justice}(x)$

Verbs

np / n

n

(np \ s) / (np \ S_{inf})

Le

parquet

peut

$\lambda y. \iota x (y \ x)$

$\lambda z. \text{parquet}(z)$

$\lambda yx. \text{pouvoir}(x, (yx))$

(np \ S_{inf})

saisir la justice

$(\lambda yx. \text{saisir}(x, y) \ \iota x. \text{justice}(x))$

Verbs

np / n

n

(np \ s) / (np \ S_{inf})

Le

parquet

peut

$\lambda y. \iota x (y \ x)$

$\lambda z. \text{parquet}(z)$

$\lambda yx. \text{pouvoir}(x, (yx))$

(np \ S_{inf})

saisir la justice

$(\lambda x. \text{saisir}(x, \iota v. \text{justice}(v)))$

Verbs

np

$(np \setminus s) / (np \setminus S_{inf})$

Le parquet

peut

$(\lambda y. \iota x (y \ x) \ \lambda z. \text{parquet}(z))$

$\lambda y x. \text{pouvoir}(x, (yx))$

$(np \setminus S_{inf})$

saisir la justice

$(\lambda x. \text{saisir}(x, \iota v. \text{justice}(v)))$

Verbs

np

Le parquet

$\iota x.(\lambda z.\text{parquet}(z) x)$

$(\text{np} \setminus s) / (\text{np} \setminus \text{Sinf})$

peut

$\lambda yx.\text{pouvoir}(x, (yx))$

$(\text{np} \setminus \text{Sinf})$

saisir la justice

$(\lambda x.\text{saisir}(x, \iota v.\text{justice}(v)))$

Verbs

np

$(np \setminus s) / (np \setminus S_{inf})$

Le parquet

peut

$\iota x. \text{parquet}(x)$

$\lambda yx. \text{pouvoir}(x, (yx))$

$(np \setminus S_{inf})$

saisir la justice

$(\lambda x. \text{saisir}(x, \iota v. \text{justice}(v)))$

Verbs

np

Le parquet

$\iota x.\text{parquet}(x)$

np \ s

peut saisir la justice

$(\lambda yx.\text{pouvoir}(x,(yx)) (\lambda x.\text{saisir}(x, \iota v.\text{justice}(v))))$

Verbs

np

Le parquet

$\iota x.\text{parquet}(x)$

np \ s

peut saisir la justice

$\lambda x.\text{pouvoir}(x, (\lambda y.\text{saisir}(y, \iota v.\text{justice}(v)) x))$

Verbs

np

Le parquet

$\iota x.\text{parquet}(x)$

np \ s

peut saisir la justice

$\lambda x.\text{pouvoir}(x,\text{saisir}(x, \iota v.\text{justice}(v)))$

Verbs

s

Le parquet peut saisir la justice

$(\lambda x.\text{pouvoir}(x,\text{saisir}(x, \iota v.\text{justice}(v)))) \iota x.\text{parquet}(x))$

Verbs

s

Le parquet peut saisir la justice

$\text{pouvoir}(\iota x.\text{parquet}(x), \text{saisir}(\iota x.\text{parquet}(x), \iota v.\text{justice}(v)))$

Control Verbs

Naomi Campbell a promis à Oprah Winfrey d'être moins violente

→ Naomi Campbell va être moins violente

→ “Naomi Campbell” is subject of “promettre” and of “être”

Teri Hatcher a permis à sa fille de rencontrer Justin Bieber

→ La fille de Teri Hatcher rencontre Justin Bieber

→ “sa fille” is objet of “promettre” and object of “rencontrer”

We assign the same formula to both “promis” and “permis”

$$((np \setminus s_{ppart}) / (np \setminus s_{deinf})) / pp_a$$

Control Verbs

We assign the same formula to both “promis” and “permis”

$$((np \setminus s_{ppart}) / (np \setminus s_{deinf})) / pp_a$$

However, the *semantics* will be different

promis: $\lambda y^e. \lambda z^{e \rightarrow t}. \lambda x^e. promettre(x, y, (z \ x))$

$promettre(Naomi, Oprah, \hat{e}tre(Naomi, moins_violente))$

permis: $\lambda y^e. \lambda z^{e \rightarrow t}. \lambda x^e. permettre(x, y, (z \ y))$

$permettre(Teri, sa_fille, rencontrer(sa_fille, Justin))$

Passives

Le niveau est fixé par les marchés

≈ les marchés fixent le niveau

“est fixé”

$(np \setminus s) / pp_{par}$

$\lambda xy. \text{fixer}(x, y)$

I abstract away from
the complications of
“être” + past participle
for the moment.

Traces

...cet equilibre delicate que l'Iran et l'Arabie saoudite
chercheront à sauvegarder ε à Genève.

... [x:B] ...

⋮

t:A

$$\begin{array}{c}
 \frac{A / \diamond \square B : \lambda x.t}{\text{l'Iran et l'Arabie saoudite}} \quad \frac{\text{chercheront à} \quad \frac{(\text{np} \setminus \text{s}_{\text{inf}}) / \text{np} \quad [\text{np}]_{[E]}}{\text{np} \setminus \text{s}_{\text{inf}}}_{[E]}}{\text{np} \setminus \text{s}}_{[E]} \\
 \frac{\text{que} \quad \frac{(\text{n} \setminus \text{n}) / (\text{s} / \diamond \square \text{np})}{\text{n} \setminus \text{n}}}{\text{s} / \diamond \square \text{np}}_{[E]} \quad \frac{\text{s}}{\text{s}}_{[I]}
 \end{array}$$

Complex Adjectives

$(n \setminus n) / pp_a$

supérieur (à)

$(n \setminus n) / pp_a$

sensibles (à)

$(n \setminus n) / pp_a$

analogue (à)

And others: conforme, favorable, nécessaire, relative, ...

$(n \setminus n) / pp_{de}$

responsable (de)

$(n \setminus n) / pp_{de}$

proche (de)

$(n \setminus n) / pp_{de}$

conscient (de)

Semantics:

$\lambda y. \lambda x. \text{supérieur_à}(x, y)$

$\lambda y. \lambda x. \text{ADJ_PRP}(x, y)$

Complex Noun Phrases

np/n n (n\n)/n n n\n n\n

Le rôle des formations politiques Ouest-allemandes

(n\n)/np np/n n (n\n)/n n

dans cette reconstitution du paysage

n\n (n\n)/n n
politique en RDA

Finding the correct bracketing is not obvious
"politiques" and "Ouest-allemandes" modify
"formations", "dans" modifies "rôle", "politique"
modifies "paysage", "en RDA" modifies "
"reconstitution".

Complex Noun Phrases

np/n n n\n (n\n)/np np/n n
L' arrivée massive sur le marché

(n\n)/n n n\n (n\n)/n n
de mètres carrés de bureaux

≈ Des mètres carrés de bureaux sont
arrivés massivement sur le marché

Complex Noun Phrases

np/n (n/pp_{de})/pp n\1n pp_{sur}/np np/n n
L' arrivée massive sur le marché

pp_{de}/n n/pp_{de} n\1n pp_{de}/n n
de mètres carrés de bureaux

≈ Des mètres carrés de bureaux sont
arrivés massivement sur le marché

Alternative analysis:
"arrivée" gets a formula
much like the past
participle "arrivé" would
and "mètres" selects for
a pp_{de} argument.
Note however that this
requires modal
decoration on the
adjectives.
Currently, this is not the
way things are
implemented.

Complex Noun Phrases

np/n	(n/pp _{de})	pp _{de} /np	np
Le	frère	de	Jean

- ◆ “frère” is semantically a binary relation: $\lambda x.\text{frère}(x,\text{Jean})$
- ◆ however, the solution to analyse “frère” and similar words (“père”, “mère”, “président”) as n/pp_{de} has some drawbacks.

Complex Noun Phrases

np/n	(n/pp _{de})	n \ ₁ n	pp _{de} /np	np
Le	frère	aîné	de	Jean

- ◆ however, the solution to analyse “frère” and similar words (“père”, “mère”, “président”) as n/pp_{de} has some drawbacks.
- ◆ we need discontinuous modes, since adjectives can intervene between “frère” and “de”

Complex Noun Phrases

np / (n / pp_{de}) (n / pp_{de})
Son frère

- ◆ however, the solution to analyse “frère” and similar words (“père”, “mère”, “président”) as n / pp_{de} has some drawbacks.
- ◆ we to update the type of the possessive article as well if we want to treat “son frère” the same way as “frère de lui”

Complex Noun Phrases

np/n

n

Son

frère

np/n

n

n\n

(n\n)/np

np

Le

frère

aîné

de

Jean

- ◆ for these reasons, the extracted lexicon does not distinguish between arguments and adjuncts inside noun phrases:
everything (except the determiner) is treated as an adjunct
- ◆ this simplifies the categories and in addition does not require any manual intervention

Adverbs

- ◆ Syntactically, adverbs behave much like prepositional phrases and they have much of the same possibilities in terms of formula assignments.
- ◆ Semantically, adverbs are quite complicated.
- ◆ But first, an example.

eg. Kamp & Reyle claim
the semantics of adverbs
is still in its infancy.
"chacun" - "tous"
"toujours"

Adverbs

np/n n (np\s)/(np\s_{inf}) (np\s)/(np\s) (np\s_{inf}) (np\s)/(np\s)

La régulation peut encore fonctionner longtemps

- ◆ Syntactically, there is not a lot going on “encore” and “longtemps” both modify the infinitive “fonctionner”
- ◆ Getting the right semantics for the adverbs is far from obvious. For the moment we will just use a very naive Davidsonian account.

Adverbs

np/n n (np\s)/(np\s_{inf}) (np\s)/(np\s) (np\s_{inf}) (np\s)/(np\s)

La régulation peut encore fonctionner longtemps

$\lambda x y e. \text{encore}(e) \wedge ((y \ x) \ e)$

$\lambda x e. \text{fonctionner}(e, x)$

$\lambda x y e. \text{longtemps}(e) \wedge ((y \ x) \ e)$

Adverbs

np/n n (np\s)/(np\s_{inf}) (np\s)/(np\s) (np\s_{inf}) (np\s)/(np\s)

La régulation peut

longtemps

Adverbs

$\lambda x y e. \text{encore}(e) \wedge ((y \ x))$

$\lambda x e. \text{fonctionner}(e, x)$

$\lambda x y e. \text{longtemps}(e) \wedge ((y \ x) \ e)$

Remark: the semantics for adjectives which is given here works only for intersective adjectives (eg. "rouge" but not "grand" or "futur"). The same is true for the adverbs: "longtemps" is in this respect similar to "grand", "encore" is still more complicated.

For a syntactic point of view, everything works well. However, as discussed in the last session, we will use $s \setminus 1 \ s$ for the adverbs instead of $(np \setminus s) / (np \setminus s)$. Since we are interested only in predicating over the event variable, this type suffices and it allows us to generalize over the adverbs appearing in different position without multiplying the formula assignments

$\lambda x. \text{voiture}(x)$

$\lambda y x. \text{rouge}(x) \wedge (y \ x)$

Complex Adverbs

$(s \setminus_1 s) / pp_a$

conformément (à)

$(s \setminus_1 s) / pp_a$

contrairement (à)

$(s \setminus_1 s) / pp_a$

quant à

Semantics:

$\lambda y. \lambda x. \lambda e. \text{conformément_à}(e, y) \wedge (x e)$

In general:

$\lambda y. \lambda x. \lambda e. \text{ADV_PRP}(e, y) \wedge (x e)$

Complex Adverbs

$(s \setminus_1 s) / pp_a$

conformément (à)

$(s \setminus_1 s) / pp_a$

contrairement (à)

$(s \setminus_1 s) / pp_a$

quant à

$(s \setminus_1 s) / pp_{de}$

lors (de)

$(s \setminus_1 s) / pp_{de}$

au-dessus (de)

$(s \setminus_1 s) / (np \setminus s_{deinf})$

$(s \setminus_1 s) / pp_{de}$

loin (de)

Adverbs

np (np \ s) / (np \ s_{ppart}) (np \ s) / (np \ s) (np \ s_{ppart}) / np np / n n
Jean a admirablement aidé sa voisine

- ◆ Deux lectures sont possibles
- ◆ Le fait que Jean a aidé sa voisine fut admirable
- ◆ La façon dont Jean a aidé sa voisine fut admirable

Adverbs

Agnelli et Worms détiennent ensemble 53.9%
des droits de vote de Saint Louis

- ◆ The adverb “ensemble” forces a collective reading for “Agnelli et Worms”
- ◆ Requires a plural sujet

Adverbs

... les deux partenaires qui en détiendront chacun 50 %

- ◆ The adverb “chacun” forces a distributive reading for “les deux partenaires”
- ◆ Requires a plural subject

Adverbs

Car les industriels, pas fous, sont tous à la recherche du gadget qui va prendre ...

- ◆ The adverb “tous” functions as a so-called “floating quantifier”
- ◆ “les industriels ... tous” \approx “tous les industriels”

Conclusions

- ◆ We can apply the ideas of Montague to treat a number of linguistically interesting phenomena.
- ◆ In many cases (especially in the case of adverbs) the solutions are very naive, but I am looking for better ways to handle them.
- ◆ Tense / aspect and comparatives are at the moment not treated at all.