On Aphasic Comprehension and Working Memory Load

Glyn Morrill [1]

Departament de Llenguatges i Sistemes Informàtics,
Universitat Politècnica de Catalunya,
Mòdul C 5 - Campus Nord,
Jordi Girona Salgado 1-3,
08034 Barcelona

Anna Gavarró [2]

Departament de Filologia Catalana,
Universitat Autònoma de Barcelona,
08193 Bellatera,
Barcelona

Abstract

There is debate in the literature on aphasia as to the extent to which aphasia arises from loss of knowledge of language vs. loss of processing resources. In this paper we compare a simple measure of the complexity of processing Type Logical Grammar with the aphasic performance observed in a large study carried out by Caplan and his colleagues. To the extent that a correlation is found, and the type logical model is taken to be psychologically plausible, our analysis suggests that aphasics suffer a deficit of working memory capacity in the incremental comprehension of language.

Key words: aphasia, type logical grammar, proof nets, memory

Email addresses: `morrill@lsi.upc.es` (Glyn Morrill), `anna.gavarro@uab.es` (Anna Gavarró).
URLs: `http: //www-lsi.upc.es/~morrill` (Glyn Morrill), `http: //seneca.uab.es/ggt/Gavarro/gavarro.html` (Anna Gavarró).
[1] Work supported by CICYT project TIC2002–04019–C03–01.
[2] Work supported by CICYT project BFF2003–08364–C02–01.

1 Introduction

According to the dependency locality theory (DLT) of [7], the resources required for storing a partially processed structure are proportional to the number of incomplete syntactic dependencies at that point in processing the structure. [11] and [15] instantiate this view by reference to proof nets as the syntactic structures of Type Logical Grammar ([14]). In type logical proof nets, dependencies are identity links and the number of unresolved syntactic dependencies at a word boundary is simply the number of identity links spanning that point. [11] argues that the maximal number of unresolved dependencies so-measured underlies the unacceptability of centre-embedding. [15] argues that the measure of the incremental course of complexity (complexity profile) of analyses underlies also processing phenomena such as garden pathing, right association (right attachment), left-to-right quantifier scoping in passives and actives, preference for passivisation of nested sentential subjects, and heavy noun phrase shift, in what we might call type logical DLT.

Complexity-based accounts are not always computationally rigorous, thus [9, p.56] writes: "Take complexity-based accounts .... Here, the idea is to establish a nonarbitrary metric for complexity, one that makes reference to structure. These metrics are rarely spelled out explicitly or motivated theoretically." On the other hand, computational rigor often requires ancillary assumptions, as noted by [17, p.1145]: "...the complexity of the component processes in sentence processing does not lend itself well to developing [computational] models that make close contact with empirical data without making numerous ancillary assumptions." Type logical DLT, we suggest, squares this circle. At the same time that it is computationally rigorous, assuming uniform cost for dependencies no ancillary assumptions are required to draw the complexity profile prediction of processing complexity. In this paper we analyse whether *aphasic* processing complexities correlate with type logical DLT complexity profiles. We find that they do, suggesting that aphasics suffer a deficit of working memory capacity in the incremental comprehension of language.

[10] describes a computational model of aphasic sentence comprehension based on the premise that all aphasics share a common deficit in the activation resources of working memory. They simulate the data from [2] (see [3]) of aphasic patients from all major syndrome types. In their system "As each word comes in, the model attempts to incorporate it as much as possible into the evolving syntactic and semantic representation. First, the word is perceptually encoded. Then, lexical access makes available its meaning and syntactic class and, in the case of verbs, also its argument structure. Based on its word class and a grammar, the word is integrated into a parse tree representation. The thematic role mapping component computes thematic-role bindings." ([10, p.88]). "The hypothesized resource reduction in aphasia was then induced by

decreasing the model's working memory capacity considerably, reducing it by half to a level of 15 activation units, to optimize the fit with the Caplan et al. data." ([10, p.96]).

Our result echoes that of [10] with a different grammar architecture, one that does not make reference to distinct syntactic structure, argument structure and thematic-role bindings.

We focus on experiment 2 reported in [3] Chapter 4. In this experiment the subjects were 37 largely unselected native English speaker aphasics. The experimenter read a sentence with normal nonemphatic intonation and the subjects were required to perform an object-manipulation task using toys to demonstrate semantic features of the sentence. The types of sentences were as follows:

(1)	Active	A	The rat hit the dog.
	Dative	D	The rat gave the dog to the cow.
	Conjoined	C	The rat hit the dog and kissed the cow.
	Passive	P	The rat was hit by the dog.
	Dative Passive	DP	The rat was given to the dog by the cow.
	Object-Subject relative	OS	The rat hit the dog that kissed the cow.
	Subject-Object relative	SO	The rat that the dog hit kissed the cow.
	Cleft-Subject	CS	It was the rat that hit the dog.
	Cleft-Object	CO	It was the rat that the dog hit.

The sentences are all unambiguous except the Object-Subject relative (OS) type, which has a right extraposed reading in which the relative clause modifies the subject. This is presumably an unintended defect in experimental design since apparent aphasic error could be the consequence of interference from the grammatical extraposition reading. (Indeed, in only this case attributed error is much higher than predicted by our model.)

2 The formalism of basic Type Logical Grammar

We present the formalism of basic Type Logical Grammar, sufficient for our current purposes, this being an embellishment of the Lambek calculus ([12]) with semantics along the lines of [13].

2.1 Prosodics

Let a *prosodic structure* be a semigroup, i.e. an algebra $(L, +)$ of arity (2) such that $+$ is associative:

(2) $s_1+(s_2+s_3) = (s_1+s_2)+s_3$

Let there be a set A of *prosodic constants*. Then the set $\mathbf{A}$ of *prosodic forms* is defined by:

(3) $\mathbf{A} ::= A \mid \mathbf{A}+\mathbf{A}$

A *prosodic interpretation* comprises a prosodic structure $(L, +)$ and a *prosodic valuation* mapping from A into L. Then the *prosodic value* $[\alpha]$ of a prosodic form α relative to a prosodic interpretation with prosodic valuation v is defined by:

(4) $[\mathbf{c}] = v(\mathbf{c})$ for prosodic constant $\mathbf{c}$

$[\alpha+\beta] = [\alpha]+[\beta]$

Prosodic forms α and β are *equivalent*, $\alpha \equiv \beta$, if and only if $[\alpha] = [\beta]$ in every prosodic interpretation. Clearly we have:

(5) $\alpha+(\beta+\gamma) \equiv (\alpha+\beta)+\gamma$

So we can omit parentheses in prosodic forms.

2.2 Semantics

Let the set $\mathbf{T}$ of *semantic types* be defined by:

(6) $\mathbf{T} ::= \mathrm{e} \mid \mathrm{t} \mid \mathbf{T} \rightarrow \mathbf{T} \mid \mathbf{T}\&\mathbf{T}$

Then the *semantic structure* induced by a non-empty set E of *entities* is the $\mathbf{T}$-indexed family of sets $\{D_\tau\}_{\tau\in\mathbf{T}}$ such that:

(7)
$$\begin{aligned} D_{\mathrm{e}} &= E \\ D_{\mathrm{o}} &= \mathbf{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\} \\ D_{\tau\to\tau'} &= D_{\tau'}^{D_\tau} \\ D_{\tau\&\tau'} &= D_\tau \times D_{\tau'} \end{aligned}$$

Let there be a set X_τ of *semantic variables* of each semantic type τ, and a set C_τ of *semantic constants* of each semantic type τ, including the *logical semantic constants*:

(8)
$$\begin{aligned} \neg &\in C_{\mathrm{t}\to\mathrm{t}} \\ \wedge, \vee, \to &\in C_{\mathrm{t}\to(\mathrm{t}\to\mathrm{t})} \\ = &\in C_{\mathrm{e}\to(\mathrm{e}\to\mathrm{t})} \\ \forall, \exists &\in C_{(\mathrm{e}\to\mathrm{t})\to\mathrm{t}} \\ \iota &\in C_{(\mathrm{e}\to\mathrm{t})\to\mathrm{e}} \end{aligned}$$

The sets Φ_τ of *semantic terms of type* τ for each semantic type τ are defined by:

(9)
$$\begin{aligned} \Phi_\tau &::= C_\tau \mid X_\tau \mid (\Phi_{\tau'\to\tau}\ \Phi_{\tau'}) \mid \pi_1\Phi_{\tau\&\tau'} \mid \pi_2\Phi_{\tau'\&\tau} \\ \Phi_{\tau\to\tau'} &::= \lambda X_\tau \Phi_{\tau'} \\ \Phi_{\tau\&\tau'} &::= (\Phi_\tau, \Phi_{\tau'}) \end{aligned}$$

We assume a convention of left-association such that $((\phi\ \psi)\ \chi)$ may be abbreviated as $(\phi\ \psi\ \chi)$. An occurrence of a semantic variable x in a semantic term is *free* if and only if it does not fall within any $\lambda x\phi$; otherwise it is *bound*. The result $\psi\{\phi/x\}$ of substituting semantic variable x (of semantic type τ) by semantic term ϕ (of semantic type τ) is the result of replacing by ϕ every free occurrence of x in ψ; the substitution $\psi\{\phi/x\}$ is *free* if and only if no variable becomes bound in the process of replacement. A *semantic form* is a semantic term containing no free variables.

A *semantic interpretation* comprises a semantic structure $\{D_\tau\}_{\tau\in\mathbf{T}}$, a *semantic assignment* g mapping from each X_τ into D_τ, and a *semantic valuation* f

mapping from each C_τ into D_τ such that:

$$
\begin{aligned}
(10) \quad f(\neg)(m) &= \overline{m}^{\{\emptyset\}} \\
f(\wedge)(m) &= m' \mapsto m \cap m' \\
f(\vee)(m) &= m' \mapsto m \cup m' \\
f(\rightarrow)(m) &= m' \mapsto \overline{m}^{\{\emptyset\}} \cup m' \\
f(=)(m) &= m' \mapsto \{\emptyset\} \text{ if } m = m' \text{ else } \emptyset \\
f(\forall)(m) &= \textstyle\bigcap_{m' \in E} m(m') \\
f(\exists)(m) &= \textstyle\bigcup_{m' \in E} m(m') \\
f(\iota)(\{m\}) &= m
\end{aligned}
$$

The *semantic value* $[\phi]^g \in D_\tau$ of a semantic term $\phi \in \Phi_\tau$ with respect to a semantic interpretation with semantic valuation f and semantic assignment g is defined by:

$$
\begin{aligned}
(11) \quad [c]^g &= f(c) \text{ for semantic constant } c \\
[x]^g &= g(x) \text{ for semantic variable } x \\
[(\phi\ \psi)]^g &= [\phi]^g([\psi]^g) && \text{functional application} \\
[\pi_1\phi]^g &= \mathbf{fst}([\phi]^g) && \text{first projection} \\
[\pi_2\phi]^g &= \mathbf{snd}([\phi]^g) && \text{second projection} \\
[\lambda x\phi]^g &= D_\tau \ni m \mapsto [\phi]^{(g-\{(x,g(x))\})\cup\{(x,m)\}} \text{ for } x \in V_\tau && \text{functional abstraction} \\
[(\phi, \psi)]^g &= \langle [\phi]^g, [\psi]^g \rangle && \text{pair formation}
\end{aligned}
$$

Semantic terms ϕ and ψ are *equivalent*, $\phi \equiv \psi$, if and only if $[\alpha]^g = [\beta]^g$ in every semantic interpretation. We have the laws of $\alpha\beta\eta$-conversion:

(12) a. $\lambda x\phi \equiv \lambda y(\phi\{y/x\})$ α-conversion

provided y is not free in ϕ and $\phi\{y/x\}$ is free

b. $(\lambda x\phi\ \psi) \equiv \phi\{\psi/x\}$ β-conversion

provided $\phi\{\psi/x\}$ is free

$\pi_1(\phi, \psi) \equiv \phi \qquad \pi_2(\phi, \psi) \equiv \psi$

c. $\lambda x(\phi\ x) \equiv \phi$ η-conversion

provided x is not free in ϕ

$(\pi_1\phi, \pi_2\phi) \equiv \phi$

We also have semantic equivalences arising in virtue of the logical semantic constants, for example:

(13) $(\wedge\ (=\ \phi\ \phi)\ \psi) \equiv \psi$

$(\exists\ \lambda x(\wedge\ \phi\ (=\ x\ \psi))) \equiv \phi\{\psi/x\}$ provided $\phi\{\psi/x\}$ is free

2.3 Syntax

Let there be a set $\mathbf{A}$ of *atomic syntactic types*. Then the set $\mathbf{F}$ of *syntactic types* is defined by:

(14) $\mathbf{F} ::= \mathbf{A} \mid \mathbf{F}{\cdot}\mathbf{F} \mid \mathbf{F}\backslash\mathbf{F} \mid \mathbf{F}/\mathbf{F}$

The operators $\cdot$, $\backslash$ and $/$ are referred to product, left division and right division respectively. Let there be a *basic type map* t mapping from $\mathbf{A}$ into $\mathbf{T}$. This induces the *type map* T from $\mathbf{F}$ into $\mathbf{T}$ such that:

$$
\begin{aligned}
(15)\ T(P) &= t(P) \text{ for atomic syntactic type } P \\
T(A{\cdot}B) &= T(A)\&T(B) \\
T(A\backslash C) &= T(A) \to T(C) \\
T(C/B) &= T(B) \to T(C)
\end{aligned}
$$

A *syntactic interpretation* comprises a prosodic structure $(L, +)$, a semantic structure $\{D\}_{\tau\in\mathbf{T}}$, and a syntactic valuation F sending each $P \in \mathbf{A}$ into a subset of $L \times D_{t(P)}$. Then the value $[\![A]\!] \subseteq L \times D_{T(A)}$ of a syntactic type with respect to a syntactic interpretation with valuation F is defined by:

$$
\begin{aligned}
(16)\ [\![P]\!] &= F(P) \text{ for atomic syntactic type } P \\
[\![A{\cdot}B]\!] &= \{(s_1{+}s_2, \langle m_1, m_2\rangle)|\ \exists(s_1, m_1) \in [\![A]\!], (s_2, m_2) \in [\![B]\!]\} \\
[\![A\backslash C]\!] &= \{(s_2, m_2)|\ \forall(s_1, m_1) \in [\![A]\!], (s_1{+}s_2, m_2(m_1)) \in [\![C]\!])\} \\
[\![C/B]\!] &= \{(s_1, m_1)|\ \forall(s_2, m_2) \in [\![B]\!], (s_1{+}s_2, m_1(m_2)) \in [\![C]\!])\}
\end{aligned}
$$

A *type assignment statement* $\alpha{-}\phi$: A comprises a syntactic type A, a prosodic form α, and a semantic form ϕ of semantic type $T(A)$. A prosodic, semantic and syntactic interpretation is a *model of a type assignment statement* $\alpha{-}\phi$*:* A if and only if $\langle[\alpha], [\phi]^g\rangle \in [\![A]\!]$; it is a *model of a set* Σ *of type assignment statements* if and only if it is a model of every type assignment statement $\sigma \in \Sigma$.

A set Σ of type assignment statements *entails* a type assignment statement σ, $\Sigma \models \sigma$, if and only if every model of Σ is also a model of σ. A *lexicon* is a set of type assignment statements. The *language* $\mathbf{L}(Lex)$ defined by a lexicon Lex is the set of type assignment statements that it entails:

(17) $\mathbf{L}(Lex) = \{\alpha{-}\phi\text{: } A|\ Lex \models \alpha{-}\phi\text{: } A\}$

and$-\lambda x\lambda y\lambda z(\wedge\ (y\ z)\ (x\ z))$: ((N\S)\(N\S))/(N\S)

by$-\lambda x\lambda y\lambda z(\wedge\ (y\ z)\ (=\ x\ z))$: ((N\S-)\(N\S-))/N

cow$-$*cow*: CN

dog$-$*dog*: CN

gave$-$*give*: ((N\S)/PP)/N

given$-(\lambda x\lambda y\lambda z(\wedge\ (y\ z)\ (\exists\ (x\ z))), give)$: ((CN\CN)/(N\(N\S-)))·(N\((N\S-)/PP))

hit$-$*hit*: (N\S)/N

hit$-(\lambda x\lambda y\lambda z(\wedge\ (y\ z)\ (\exists\ (x\ z))), hit)$: (((CN\CN)/(N\(N\S-)))·(N\(N\S-))

it$-\lambda xx$: NPit

kissed$-$*kiss*: (N\S)/N

rat$-$*rat*: CN

that$-\lambda x\lambda y\lambda z(\wedge\ (y\ z)\ (x\ z))$: (CN\CN)/(N\S)

that$-\lambda x\lambda y\lambda z(\wedge\ (y\ z)\ (x\ z))$: (CN\CN)/(S/N)

the$-\iota$: N/CN

to$-\lambda xx$: PP/N

was$-\lambda x\lambda y(x\ (=\ y)\ y)$: (N\S)/(CN\CN)

was$-\lambda x\lambda y\lambda z(z\ (y\ (=\ x)\ x))$: ((NPit\S)/(CN\CN))/N

Fig. 1. Type logical lexicon

3 Grammar

Let there be the following basic types and the lexicon in figure 1.

(18)

Atomic syntactic type	P	$t(P)$
count noun	CN	e→t
proper name	N	e
expletive pronoun	NPit	t→t
prepositional phrase	PP	e
declarative sentence	S	t
abstract passive sentence	S-	t

Then the language model defined includes the following:

(19) A **the+rat+hit+the+dog**−

$(hit\ (\iota\ dog)\ (\iota\ rat))$: S

D **the+rat+gave+the+dog+to+the+cow**−

$(give\ (\iota\ dog)\ (\iota\ cow)\ (\iota\ rat))$: S

C **the+rat+hit+the+dog+and+kissed+the+cow**−

$(\wedge\ (hit\ (\iota\ dog)\ (\iota\ rat))\ (kiss\ (\iota\ cow)\ (\iota\ rat)))$: S

P **the+rat+was+hit+by+the+dog**−

$(hit\ (\iota\ rat)\ (\iota\ dog))$: S

DP **the+rat+was+given+to+the+dog+by+the+cow**−

$(give\ (\iota\ rat)\ (\iota\ dog)\ (\iota\ cow))$: S

OS **the+rat+hit+the+dog+that+kissed+the+cow**−

$(hit\ (\iota\ \lambda z(\wedge\ (dog\ z)\ (kiss\ (\iota\ cow)\ z)))\ (\iota\ rat))$: S

SO **the+rat+that+the+dog+hit+kissed+the+cow**−

$(kiss\ (\iota\ cow)\ (\iota\ \lambda z(\wedge\ (rat\ z)\ (hit\ z\ (\iota\ dog)))))$: S

CS **it+was+the+rat+that+hit+the+dog**−

$(hit\ (\iota\ dog)\ (\iota\ rat))$: S

CO **it+was+the+rat+that+the+dog+hit**−

$(hit\ (\iota\ rat)\ (\iota\ dog))$: S

The derivations, or syntactic structures, of TLG are (proof) nets ([8], [16]), which we define in the next section.

4 Nets for basic Type Logical Grammar

A *label* is a syntactic type together with a polarity input ($^\bullet$) or output ($^\circ$). Where p is a polarity, $\overline{p}$ is the opposite polarity. Labels A^p and $A^{\overline{p}}$ are *complementary*. A *literal* is a label the type of which is atomic.

A *link* has a list of *premise* labels (above) and a list of *conclusion* labels

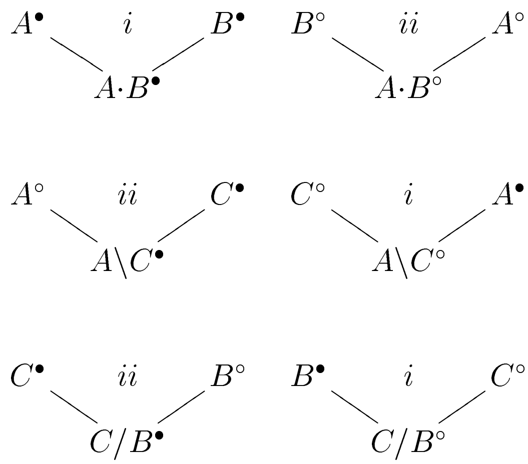
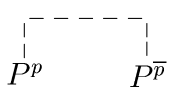


Fig. 2. Logical links

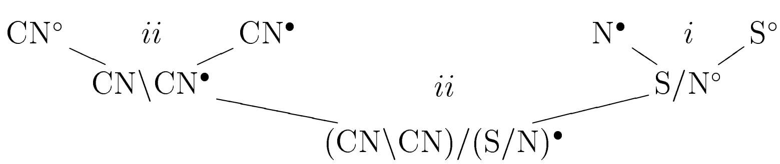
(below) with some edges between them. An *identity link* is of the form:

(20)



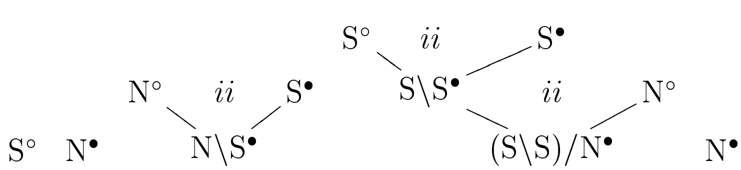
A *logical link* is one of the local trees given in Figure 2. A *polar type tree* is a tree the leaves of which are literals and each local tree of which is a logical link. An *input polar type tree* is one the root of which has input polarity; an *output polar type tree* is one the root of which has output polarity. Each label is the root of a unique polar type tree, which is the result of unfolding the label upwards according to the logical links. For example, the (input) polar type tree for $(\mathrm{CN}\backslash\mathrm{CN})/(\mathrm{S}/\mathrm{N})^\bullet$ is:

(21)



A *frame* is a list comprising an output polar type tree followed by some input polar type trees. For example, the following is a frame:

(22)



A *net* is the result of connecting by an identity link every leaf in a frame with

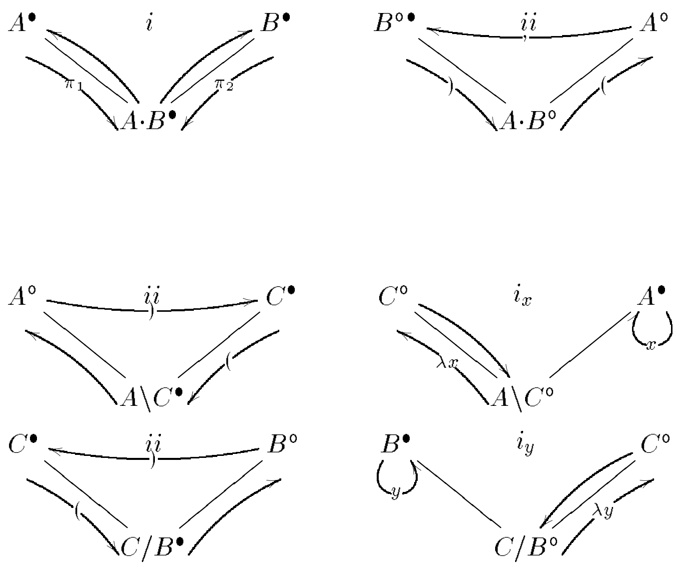


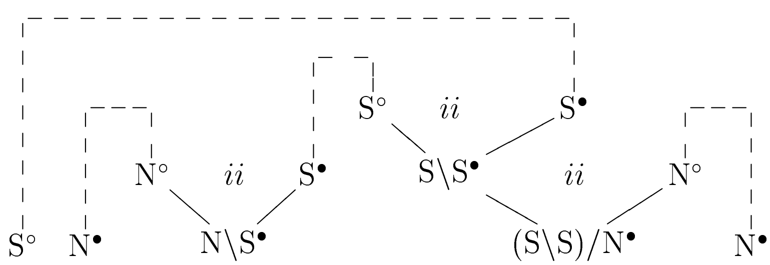
Fig. 3. Semantic trip travel instructions

a unique complementary leaf such that:

(23)	*planarity*	The identity links are planar in the list ordering.
	acyclicity	Every cycle crosses both edges of some i-link.
	no subtending	No identity link connects the leftmost and rightmost descendent leaves of an output division node.

(For planarity see [16] and [1]; for acyclicity see [4]; for no subtending see [6].) For example, the following is a net:

(24)



Let there be semantic forms of the appropriate type associated with each input polar tree in a net. The *semantic trip* of the net with each output division logical link assigned a distinct semantic variable is the trip which originates upwards at the unique output root and generates a textual form proceeding as shown in Figure 3 and bouncing with the associated semantic form at input roots ([5]). The semantic trip ends when it returns downwards at the origin. The *reading* ϕ_Π of a net Π is the textual form generated by its semantic trip.

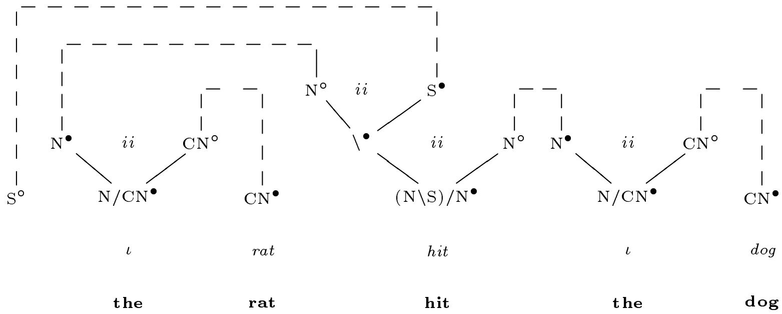


Fig. 4. Net for an Active (A) sentence

Then we have the following:

Proposition 1 (Soundness). *If* $\alpha_1 - \phi_1 \colon A_1, \ldots, \alpha_n - \phi_n \colon A_n \in Lex$ *then for every net* Π *over the frame* $A_0{}^{\circ}, \overset{A_1{}^{\bullet}}{\phi_1}, \ldots, \overset{A_n{}^{\bullet}}{\phi_n}$, $\alpha_1 + \cdots + \alpha_n - \phi_\Pi \colon A_0 \in \mathbf{L}(Lex)$.

Proposition 2 (Completeness). *If* $\alpha - \phi \colon A_0 \in \mathbf{L}(Lex)$ *then there exist* $\alpha_1 - \phi_1 \colon A_1, \ldots, \alpha_n - \phi_n \colon A_n \in Lex$ *and a net* Π *over the frame* $A_0{}^{\circ}, \overset{A_1{}^{\bullet}}{\phi_1}, \ldots, \overset{A_n{}^{\bullet}}{\phi_n}$ *such that* $\alpha \equiv \alpha_1 + \cdots + \alpha_n$ *and* $\phi \equiv \phi_\Pi$.

5 Sentences

In the following subsections we give the nets and semantics for each of the sentence types. We also give the complexity profiles, these being a plot of the number of unbounded dependencies (overarching identity links) at each word boundary. Following Johnson (1988) we refer to this number as the 'cut' and we note maximal cuts and average cuts.

5.1 Active (A)

The net for a sentence of this type is given in figure 4. The result of the semantic trip is the following:

(25) $(hit\ (\iota\ dog)\ (\iota\ rat))$

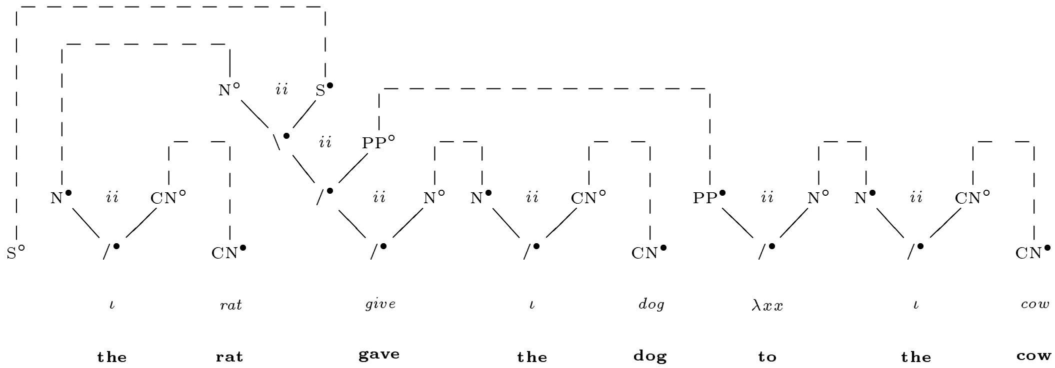


Fig. 5. Net for a Dative (D) sentence

The complexity profile is thus:

(26)

3		a				
2			a			
1	a			a	a	
0						a
	The	rat	hit	the	dog.	

The maximal cut is 3; the average cut is 1.33.

5.2 Dative (D)

The net for a sentence of this type is given in figure 5. The result of the semantic trip is (27a) which normalizes to (27b).

(27) a. $(give\ (\iota\ dog)\ (\lambda xx\ (\iota\ cow)\ (\iota\ rat))$

b. $(give\ (\iota\ dog)\ (\iota\ cow)\ (\iota\ rat))$

The complexity profile is as follows:

(28)

3		a							
2			a	a	a				
1	a					a	a	a	
0									a
	The	rat	gave	the	dog	to	the	cow.	

The maximal cut is 3; the average cut is 1.44.

5.3 *Conjoined (C)*

The net for a sentence of this type is given in figure 6. The normalized semantics is as follows.

(29) ($\wedge$ (*hit* (ι *dog*) (ι *rat*)) (*kiss* (ι *cow*) (ι *rat*)))

The complexity profile is thus:

(30)	5				a	a					
	4						a				
	3		a								
	2			a				a			
	1	a							a	a	
	0										a
		The	rat	hit	the	dog	and	kissed	the	cow.	

The maximal cut is 5; the average cut is 2.40.

5.4 *Passive (P)*

The net for a sentence of this type is given in figure 7; the analysis of passive is that of [15]. The normalized semantics is (31a) which is logically equivalent to (31b).

(31) a. ($\wedge$ (= (ι *rat*) (ι *rat*)) ($\exists$ λz($\wedge$ (*hit* (ι *rat*) z) (= (ι *dog*) z))))

b. (*hit* (ι *rat*) (ι *dog*))

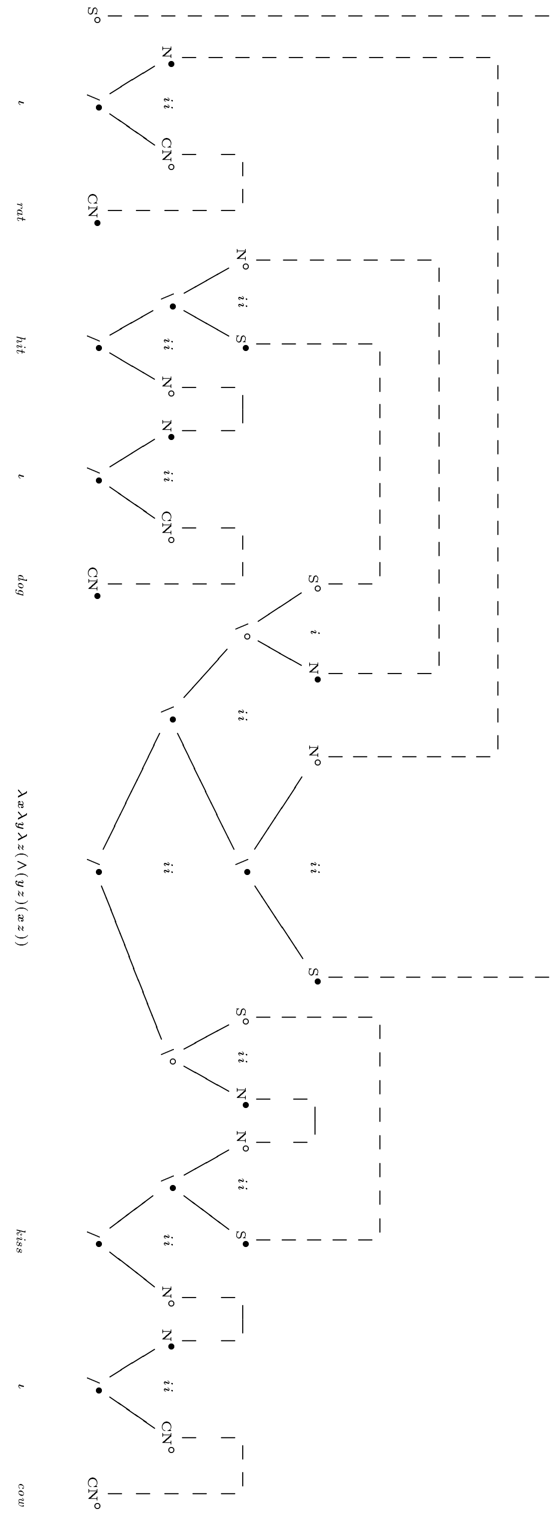


Fig. 6. Net for a Conjoined (C) sentence

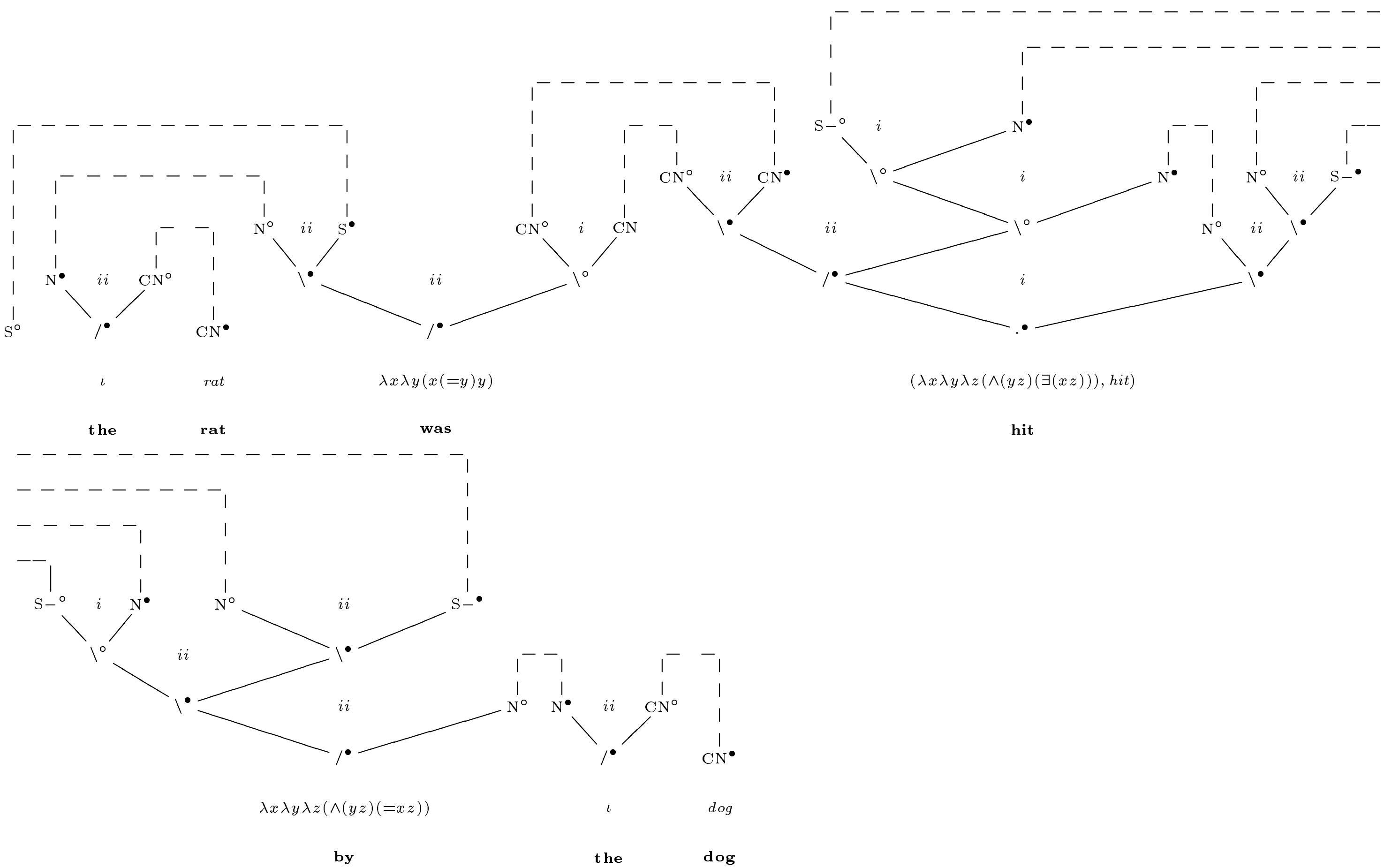


Fig. 7. Net for a Passive (P) sentence

The complexity profile is as follows:

(32)

	The	rat	was	hit	by	the	dog.	
4					a			
3		a						
2			a	a				
1	a					a	a	
0								a

The maximal cut is 4; the average cut is 1.75.

5.5 Dative Passive (DP)

The net for a sentence of this type is given in figures 8 and 9; the analysis of dative passive is a generalization of that of passive of [15]. The normalized semantics is (33a) which is logically equivalent to (33b).

(33) a. $(\wedge\ (=\ (\iota\ rat)\ (\iota\ rat))\ (\exists\ \lambda z(\wedge\ (give\ (\iota\ rat)\ (\iota\ dog)\ z)\ (=\ (\iota\ cow)\ z))))$

b. $(give\ (\iota\ rat)\ (\iota\ dog)\ (\iota\ cow))$

The complexity profile is as follows:

(34)

	The	rat	was	given	to	the	dog	by	the	cow.	
5					a	a	a				
4								a			
3		a									
2			a	a							
1	a								a	a	
0											a

The maximal cut is 5; the average cut is 2.64.

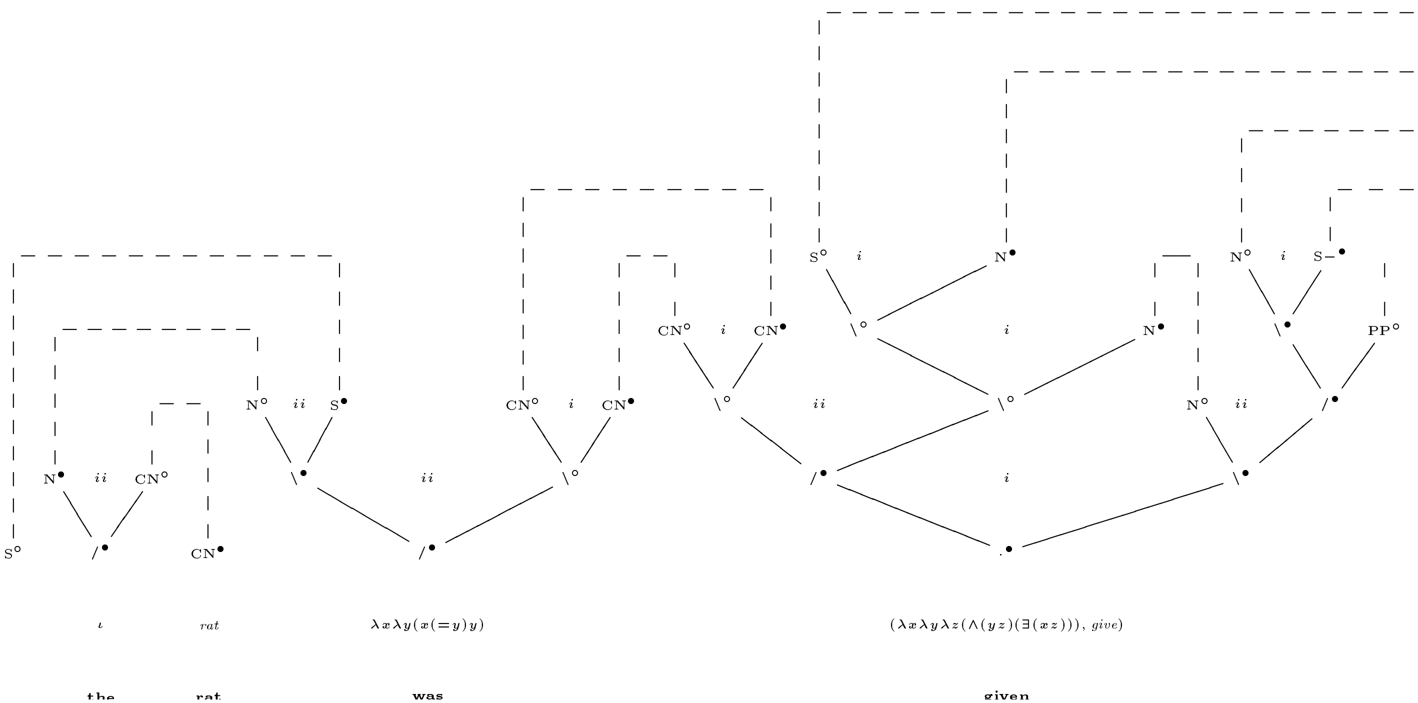


Fig. 8. Net for a Dative Passive (DP) sentence, part I

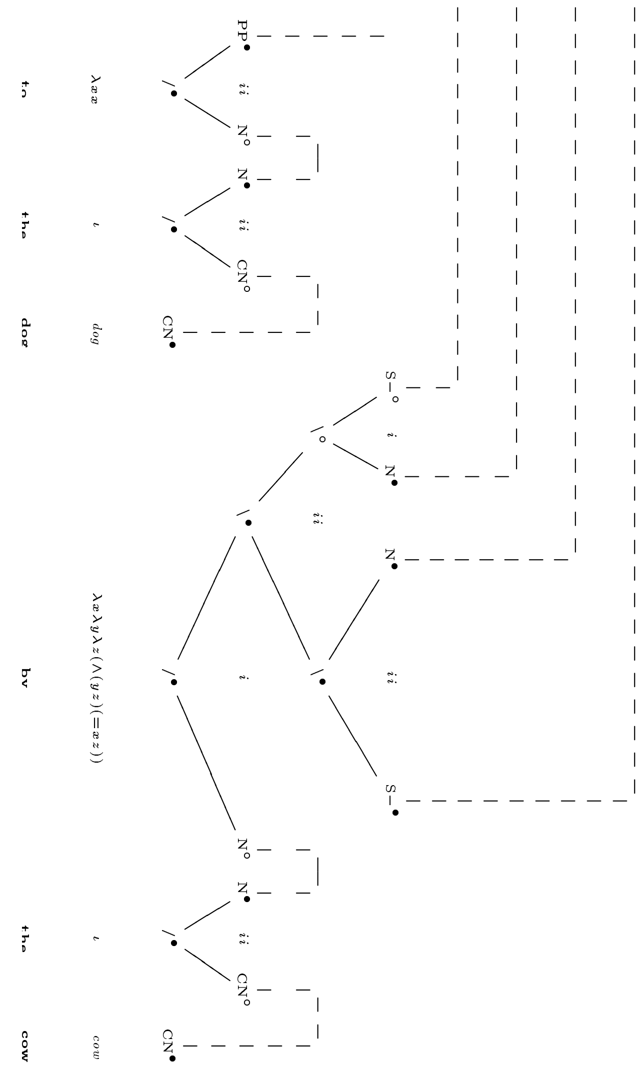


Fig. 9. Net for a Dative Passive (DP) sentence, part II

5.6 *Object-Subject relative (OS)*

The net for a sentence of this type is given in figure 10. The normalized semantics is as follows:

(35) $(hit\ (\iota\ \lambda z(\wedge\ (dog\ z)\ (kiss\ (\iota\ cow)\ z)))\ (\iota\ rat))$

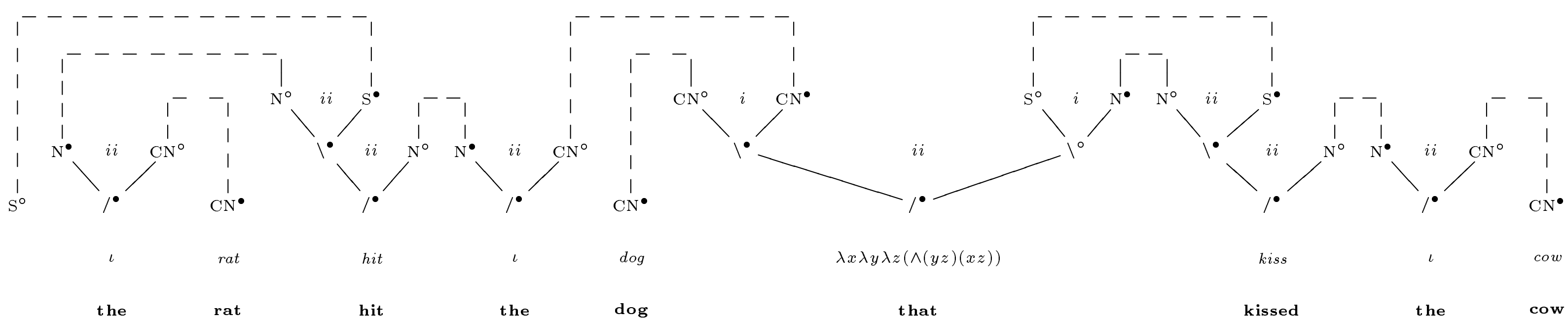


Fig. 10. Net for an Object-Subject relative (OS) sentence

The complexity profile is thus:

(36)

	The	rat	hit	the	dog	that	kissed	the	cow.	
3		a								
2			a			a	a			
1	a			a	a			a	a	
0										a

The maximal cut is 3; the average cut is 1.40.

5.7 Subject-Object relative (SO)

The net for a sentence of this type is given in figure 11. The normalized semantics is as follows:

(37) $(kiss\ (\iota\ cow)\ (\iota\ \lambda z(\wedge\ (rat\ z)\ (hit\ z\ (\iota\ dog)))))$

The complexity profile is thus:

(38)

	The	rat	that	the	dog	hit	kissed	the	cow.	
6					a					
5						a				
4			a	a						
3		a								
2							a			
1	a							a	a	
0										a

The maximal cut is 6; the average cut is 2.70.

5.8 Cleft-Subject (CS)

The net for a sentence of this type is given in figure 12; the analysis of the expletive subject of cleft constructions would appear to be new to this paper.

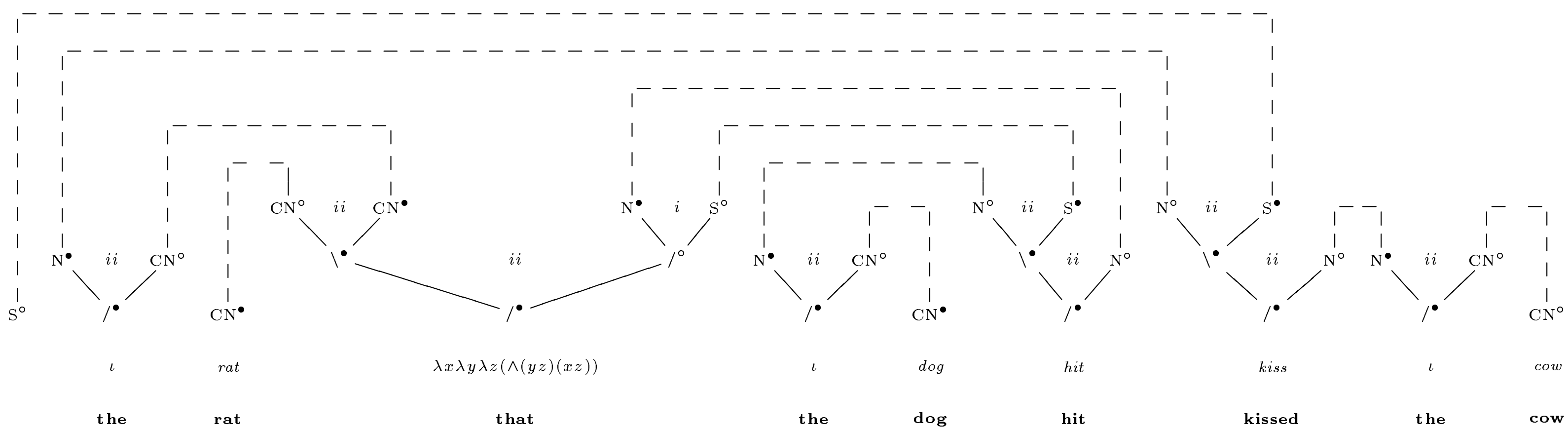


Fig. 11. Net for a Subject-Object relative (SO) sentence

The normalized semantics is (39a) which is logically equivalent to (39b).

(39) a. $(\wedge\ (=\ (\iota\ \mathit{rat})\ (\iota\ \mathit{rat}))\ (\mathit{hit}\ (\iota\ \mathit{dog})\ (\iota\ \mathit{rat})))$

b. $(\mathit{hit}\ (\iota\ \mathit{dog})\ (\iota\ \mathit{rat}))$

The complexity profile is as follows:

(40)

3			a	a					
2		a			a	a			
1	a						a	a	
0									a
	It	was	the	rat	that	hit	the	dog.	

The maximal cut is 3; the average cut is 1.67.

5.9 *Cleft-Object (CO)*

The net for a sentence of this type is given in figure 13. The normalised semantics is (41a) which is logically equivalent to (41b).

(41) a. $(\wedge\ (=\ (\iota\ \mathit{rat})\ (\iota\ \mathit{rat}))\ (\mathit{hit}\ (\iota\ \mathit{dog})\ (\iota\ \mathit{rat})))$

b. $(\mathit{hit}\ (\iota\ \mathit{dog})\ (\iota\ \mathit{rat}))$

The complexity profile is as follows:

(42)

4							a		
3			a	a				a	
2		a			a	a			
1	a								
0									a
	It	was	the	rat	that	the	dog	hit.	

The maximal cut is 4; the average cut is 2.22.

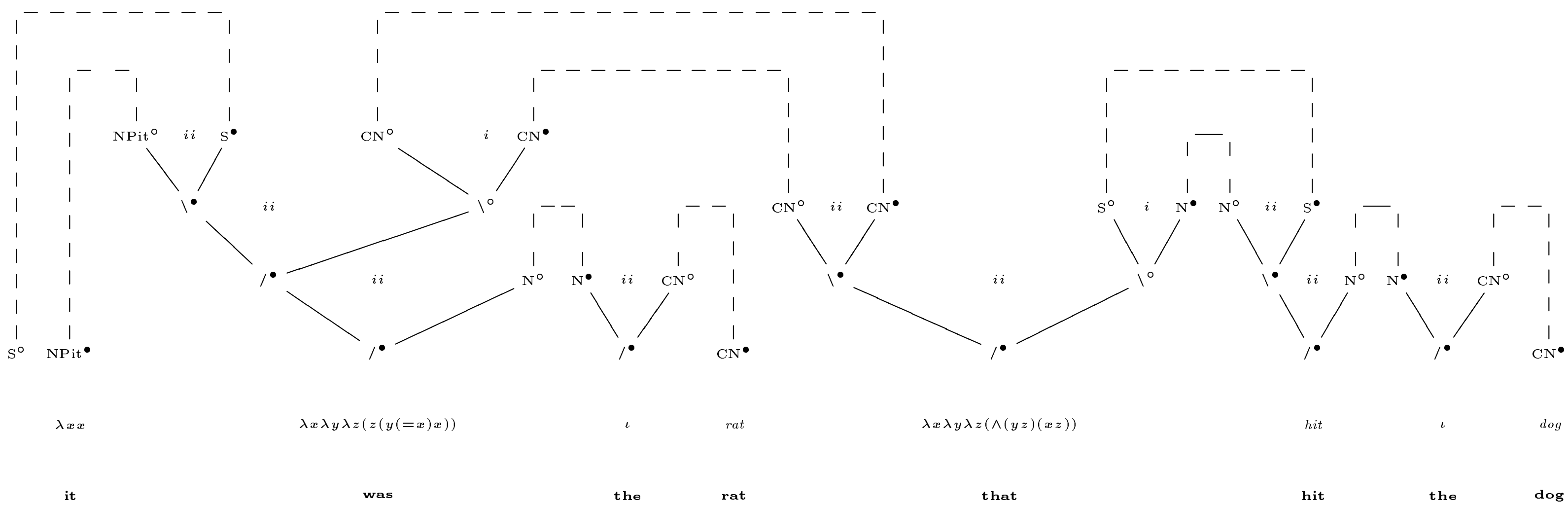


Fig. 12. Net for a Cleft-Subject (CS) sentence

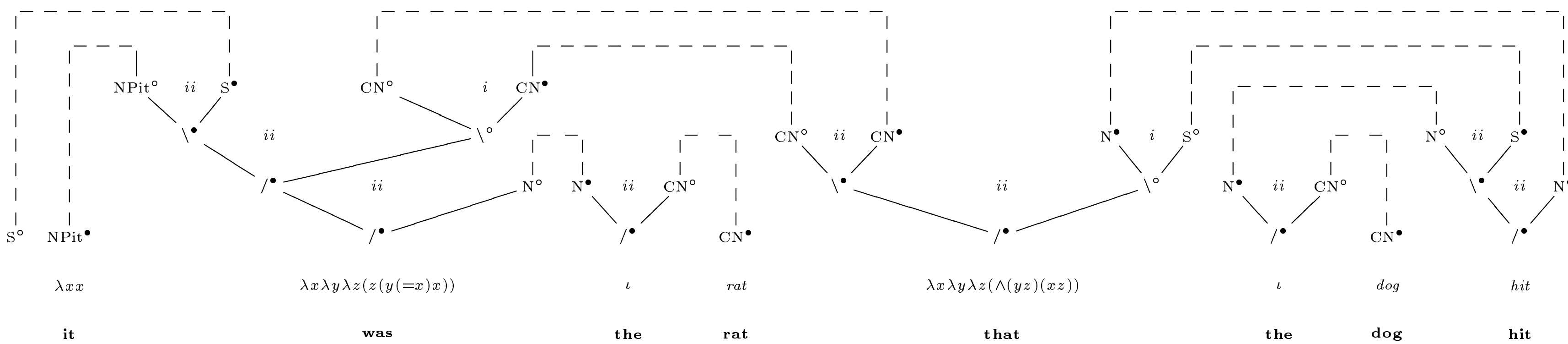


Fig. 13. Net for a Cleft-Object (CO) sentence

6 Results

The mean comprehension results on a scale of 0 (worst) to 5 (best) were as follows ([3, p.105]), compared to the maximal cut and average cut of the proof net analyses:

(43)

Sentence type	mean comprehension	maximal cut	average cut
A	4.4	3	1.33
CS	4.2	3	1.67
D	3.2	3	1.44
P	2.9	4	1.75
C	2.7	5	2.40
CO	2.6	4	2.22
OS	2.3	3	1.40
DP	2.0	5	2.64
SO	1.3	6	2.70

The maximal cut and average cut correlate quite well with the mean comprehension with the exception of the OS sentence type. As we noted in the introduction, this sentence type is ambiguous, allowing a right extraposed reading, so that the existence of this reading would be expected to detract from the prescribed interpretation. Otherwise, in particular, the A sentence type has the best comprehension and the lowest cut complexity and the SO sentence type has the worst comprehension and the highest cut complexity.

7 Conclusion

We argue that the aphasic comprehension of the sentence types of the Caplan et al. experiment correlates with the proof net complexity of these sentence types. The measure of complexity with which we correlate aphasic comprehension is an immediate reflection of grammar, no ancilliary assumption is made beyond that that dependencies are of uniform cost.

References

[1] V. Michele Abrusci. Noncommutative Proof Nets. In Yves Lafont, Jean-Yves Girard, and Laurent Regnier, editors, *Advances in Linear Logic*, London Mathematical Society Lecture Note Series 222, pages 271–296. Cambridge University Press, Cambridge, 1995.

[2] D. Caplan, C. Baker, and F. Dehaut. Syntactic determinants of sentence comprehension in aphasia. *Cognition*, 21: 117–175, 1985.

[3] David Caplan and Nancy Hildecrandt. *Disorders of Syntactic Comprehension*. The MIT Press, Cambridge, Massachusetts, 1988.

[4] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, **28**: 181–203, 1989.

[5] Philippe de Groote and Christian Retoré. On the Semantic Readings of Proof-Nets. In Geert-Jan Kruijff, Glyn Morrill, and Richard T. Oehrle, editors, *Proceedings of Formal Grammar 1996*, pages 57–70, Prague, 1996.

[6] Philippe de Groote and Christian Retoré. *Proof-theoretic methods in computational linguistics*. Lecture notes 15th European Summer School in Logic, Language and Information, Vienna, 2003.

[7] Edward Gibson. Linguistic complexity: locality of syntactic dependencies. *Cognition*, **68**:1–76, 1988.

[8] Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, **50**: 1–102, 1987.

[9] Yosef Grodzinsky. The neurology of syntax: Language use without Broca's area. *Behavioural and Brain Sciences*, **23**:1–71, 2000.

[10] Henk J. Haarmann, Marcel Adam Just, and Patricia A. Carpenter. Aphasic Sentence Comprehension as a Resource Deficit: A Computational Approach. *Brain and Language*, 59: 76–100, 1997.

[11] M. Johnson. Proof nets and the complexity of processing center-embedded constructions. *Journal of Logic, Language and Information*, 7(4):433–447, 1988.

[12] J. Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, **65**: 154–170, 1958.

[13] J. Lambek. Categorial and Categorical Grammars. In Richard T. Oehrle, Emmon Bach and Deidre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, Studies in Linguistics and Philosophy Volume 32, pages 297–317. D. Reidel, Dordrecht, 1988.

[14] Glyn Morrill. *Type Logical Grammar: Categorial Logic of Signs*. Kluwer Academic Publishers, Dordrecht, 1994.

[15] Glyn Morrill. Incremental Processing and Acceptability. *Computational Linguistics*, 26(3), 319–338. 2000.

[16] Dirk Roorda. *Resource Logics: Proof-Theoretical Investigations.* PhD thesis, Universiteit van Amsterdam, 1991.

[17] Michael K. Tanenhaus. Sentence Processing. In Lynn Nadal, editor, *Encyclopedia of Cognitive Science, Volume 3*, pages 1142–1148. Nature Publishing Group, Macmillan, London, 2003.