Combinatory Categorial Grammar

Vera Demberg

Cluster of Excellence “Multimodal Computing and Interaction”
Universität des Saarlandes

Incremental Syntax and Semantics Seminar WS 2012/13

November 14th, 2012

Table of Contents

Table of Contents

What can CCG do?

CCG = Combinatory Categorial Grammar

Selling points of CCG

- direct syntax-semantics interface
- can model many phenomena of natural language
  - long distance dependencies (relative clauses, wh-questions)
  - binding (reflexives)
  - control (object controls comp subject, e.g. *persuade*)
  - coordination
  - cross-serial dependencies (Dutch; $\rightarrow$ mildly context-free)

Properties of Combinatory Categorial Grammar

Categories:

- categories specify valency of words or constituents
- lexicon: each word is associated with one or more categories

Example

simple	categories	complex	categories
Peter	NP	a	NP/N
dog	N	sleeps	$S\backslash NP$

Operations:

- small set of combinatory rules
- can differ by language

Example

Forward Application:	X/Y	Y	$\Rightarrow_{>}$	X
Backward Application:	Y	$X\backslash Y$	$\Rightarrow_{<}$	X

Properties of **Combinatory Categorial** Grammar

Categories:

- categories specify valency of words or constituents
- lexicon: each word is associated with one or more categories

Example

simple	categories	complex	categories
Peter	NP	a	NP/N
dog	N	sleeps	$S\backslash NP$

Operations:

- small set of combinatory rules
- can differ by language

Example

Forward Application:	X/Y	Y	$\Rightarrow_{>}$	X
Backward Application:	Y	$X\backslash Y$	$\Rightarrow_{<}$	X

Some simple example derivations in CCG:

Example Lexicon

simple	categories	complex	categories
Peter	NP	a	NP/N
dog	N	sleeps	$S\backslash NP$
		saw	$(S\backslash NP)/NP$
		tired	N/N
		often	$(S\backslash NP)\backslash(S\backslash NP)$

Rules

Forward Application:	X/Y	Y	$\Rightarrow_{>}$	X
Backward Application:	Y	$X\backslash Y$	$\Rightarrow_{<}$	X

Let's derive: *A dog sleeps.*

Some simple example derivations in CCG:

Example Lexicon

simple	categories	complex	categories
Peter	NP	a	NP/N
dog	N	sleeps	$S\backslash NP$
		saw	$(S\backslash NP)/NP$
		tired	N/N
		often	$(S\backslash NP)\backslash(S\backslash NP)$

Rules

Forward Application:	X/Y	Y	$\Rightarrow_{>}$	X
Backward Application:	Y	$X\backslash Y$	$\Rightarrow_{<}$	X

Let's derive: *Peter saw a dog.*

Some simple example derivations in CCG:

Example Lexicon

simple	categories	complex	categories
Peter	NP	a	NP/N
dog	N	sleeps	$S\backslash NP$
		saw	$(S\backslash NP)/NP$
		tired	N/N
		often	$(S\backslash NP)\backslash(S\backslash NP)$

Rules

Forward Application:	X/Y	Y	$\Rightarrow_{>}$	X
Backward Application:	Y	$X\backslash Y$	$\Rightarrow_{<}$	X

Let's derive: *A tired dog sleeps.*

Some simple example derivations in CCG:

Example Lexicon

simple	categories	complex	categories
Peter	NP	a	NP/N
dog	N	sleeps	$S\backslash NP$
		saw	$(S\backslash NP)/NP$
		tired	N/N
		often	$(S\backslash NP)\backslash(S\backslash NP)$

Rules

Forward Application:	X/Y	Y	$\Rightarrow_{>}$	X
Backward Application:	Y	$X\backslash Y$	$\Rightarrow_{<}$	X

Let's derive: *Peter sleeps often.*

Table of Contents

CCG Combinatory Rules

Forward and Backward Application (with Semantics):

Forward Application: $X/Y : f \quad Y : a \quad \Rightarrow_{>} \quad X : fa$

Backward Application: $Y : a \quad X\backslash Y : f \quad \Rightarrow_{<} \quad X : fa$

Example

$$\dfrac{\dfrac{\text{Marcel}}{NP_{3sm} : marcel'} \quad \dfrac{\dfrac{\text{proved}}{(S\backslash NP_{3s})/NP : \lambda x\lambda y.prove' xy} \quad \dfrac{\text{completeness}}{NP : completeness'}}{S\backslash NP_{3s} : \lambda y.prove' completeness' y}>}{S : prove' completeness' marcel'}<$$

CCG Combinatory Rules

CCG includes further operations in order to adequately handle English.

Coordination: $X \quad (X\backslash X)/X \quad X \quad \Rightarrow_\phi \quad X$

Example

$$\dfrac{\dfrac{\dfrac{\text{Marcel}}{NP} \quad \dfrac{\dfrac{\text{conjectured}}{(S\backslash NP)/NP} \quad \dfrac{\dfrac{\dfrac{\text{and}}{(X\backslash_\star X)/_\star X} \quad \dfrac{\text{proved}}{(S\backslash NP)/NP}}{((S\backslash NP)/NP)\backslash_\star((S\backslash NP)/NP)}>}{(S\backslash NP)/NP}<}{S\backslash NP}> \quad \dfrac{\text{completeness}}{NP}}{S}<$$

CCG Combinatory Rules

Forward Composition: $X/Y : f \quad Y/Z : g \Rightarrow_{>B} X/Z : \lambda z.f(gz)$

Backward Composition: $Y\backslash Z : g \quad X\backslash Y : f \Rightarrow_{<B} X\backslash Z : \lambda z.f(gz)$

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Marcel}}{NP : marcel'} \quad \dfrac{\text{conjectured}}{(S\backslash NP)/NP : conjecture'} \quad \dfrac{\dfrac{\text{and}}{(X\backslash_\star X)/_\star X : and'} \quad \dfrac{\dfrac{\text{might}}{(S\backslash NP)/VP : might'} \quad \dfrac{\text{prove}}{VP/NP : prove'}}{(S\backslash NP)/NP : \lambda x\lambda y.might'\,(prove'\,x)\,y}\,>\mathbf{B}}{((S\backslash NP)/NP)\backslash_\star((S\backslash NP)/NP) : \lambda tv\lambda x\lambda y.and'\,(might'\,(prove'\,x)\,y)\,(tv\,xy)}\,>}{(S\backslash NP)/NP : \lambda x\lambda y.and'\,(might'\,(prove'\,x)\,y)\,(conjecture'\,xy)}\,<}{S\backslash NP : \lambda y.and'\,(might'\,(prove'\,completeness')\,y)\,(conjecture'\,completeness'\,y)}\,>}{S : and'\,(might'\,(prove'\,completeness')\,marcel')\,(conjecture'\,completeness'\,marcel')}\,<$$

CCG Combinatory Rules

Forward Generalized Composition: $X/Y \quad (Y/Z)/\$_1 \quad \Rightarrow_{>B^n} \quad (X/Z)/\$_1$

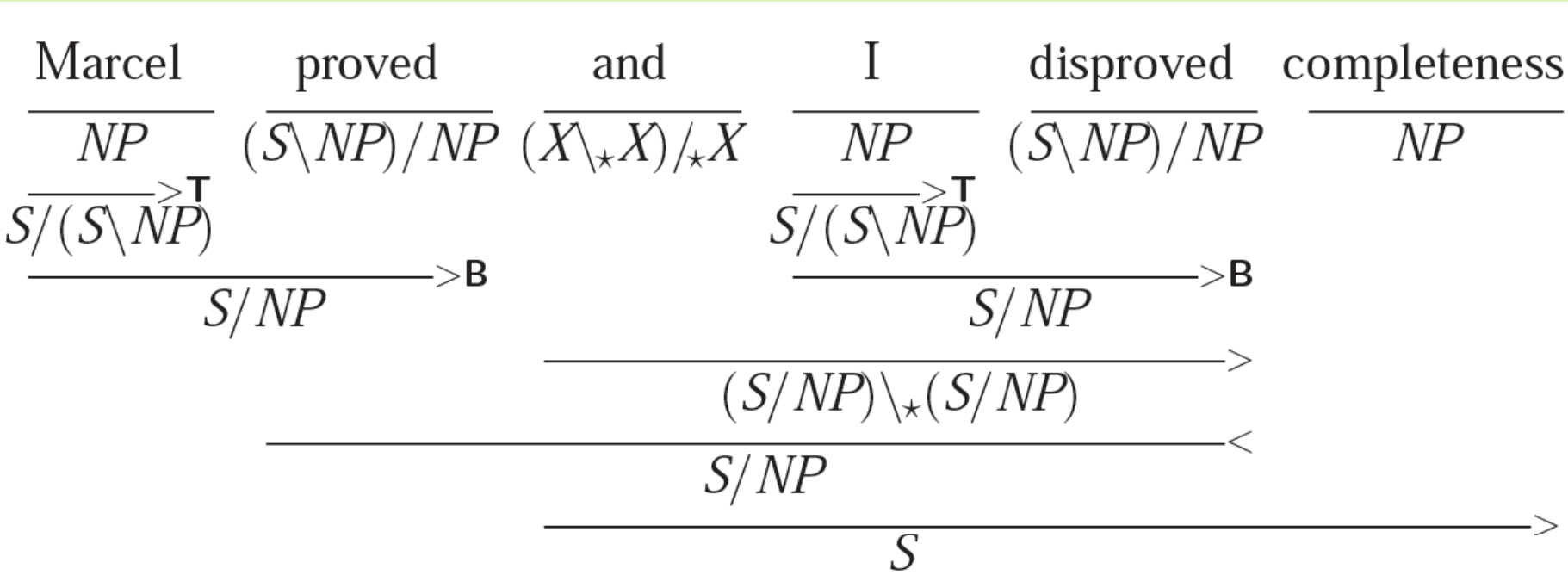
Example

$$\dfrac{\dfrac{\text{might}}{(S\backslash NP)/VP} \quad \dfrac{\text{give}}{(VP/NP)/NP}}{((S\backslash NP)/NP)/NP}\,{>}\mathbf{B}^2$$

CCG Combinatory Rules

Forward Type-raising: $X : a \quad \Rightarrow_T \quad T/(T\backslash X) : \lambda f.fa$

Example



All rules for English

Forward Application:	X/Y	Y	$\Rightarrow_{>}$	X
Backward Application:	Y	$X\backslash Y$	$\Rightarrow_{<}$	X
Forward Composition:	X/Y	Y/Z	$\Rightarrow_{>B}$	X/Z
Backward Composition:	$Y\backslash Z$	$X\backslash Y$	$\Rightarrow_{<B}$	$X\backslash Z$
Forward Generalized Composition:	X/Y	$(Y/Z)/\$_1$	$\Rightarrow_{>B^n}$	$(X/Z)/\$_1$
Backward Crossed Composition:	$Y\backslash Z$	$X\backslash Y$	$\Rightarrow_{<B_x}$	X/Z
Forward Type-raising:	X		$\Rightarrow_{T}$	$T/(T\backslash X)$
Coordination:	X conj	X	$\Rightarrow_{\phi}$	X

Table of Contents

The Strict Competence Hypothesis

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

Competence

Competence is the **'ideal' language system** that makes it possible for speakers to produce and understand an infinite number of sentences in their language, and to distinguish grammatical sentences from ungrammatical sentences.

Performance

Linguistic performance is governed by **principles of cognitive structure** such as memory limitations, distractions, shifts of attention and interest, and (random or characteristic) errors.

The Strict Competence Hypothesis

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

Competence

Competence is the **'ideal' language system** that makes it possible for speakers to produce and understand an infinite number of sentences in their language, and to distinguish grammatical sentences from ungrammatical sentences.

Performance

Linguistic performance is governed by **principles of cognitive structure** such as memory limitations, distractions, shifts of attention and interest, and (random or characteristic) errors.

The Strict Competence Hypothesis

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

+

Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.
($\Rightarrow$ entities combined by syntactic rules must be semantically interpretable)

=

Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

The Strict Competence Hypothesis

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

+

Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.
($\Rightarrow$ entities combined by syntactic rules must be semantically interpretable)

=

Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

The Strict Competence Hypothesis

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

$+$

Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.
($\Rightarrow$ entities combined by syntactic rules must be semantically interpretable)

$=$

Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

Constituents in CCG

CCG has flexible constituency structure.

Spurious ambiguity

There are 24 different ways of deriving:

Peter caught a big cat.

but they all lead to same semantic interpretation.

From the point of view of incrementality, that's great news!

Constituents in CCG

CCG has flexible constituency structure.

Spurious ambiguity

There are 24 different ways of deriving:

Peter caught a big cat.

but they all lead to same semantic interpretation.

From the point of view of incrementality, that's great news!

CCG and incrementality

Example (following pattern of *The horse raced past the barn fell*.)

a) The doctor sent for the patients arrived. (more difficult)

b) The flowers sent for the patients arrived. (less difficult)

- If b) is easier, this indicates that the processor has figured out at the point of "sent" that *flowers* cannot be the agent of a sending action.
  $\Rightarrow$ **Incremental interpretation at "the flowers sent"**
- "the flowers sent" is a CCG constituent
- Therefore, CCG can happily explain why humans prefer b).

CCG and incrementality

Example (following pattern of *The horse raced past the barn fell*.)

a) The doctor sent for the patients arrived. (more difficult)

b) The flowers sent for the patients arrived. (less difficult)

- If b) is easier, this indicates that the processor has figured out at the point of "sent" that *flowers* cannot be the agent of a sending action.
  $\Rightarrow$ **Incremental interpretation at "the flowers sent"**
- "the flowers sent" is a CCG constituent
- Therefore, CCG can happily explain why humans prefer b).

So how does that work?

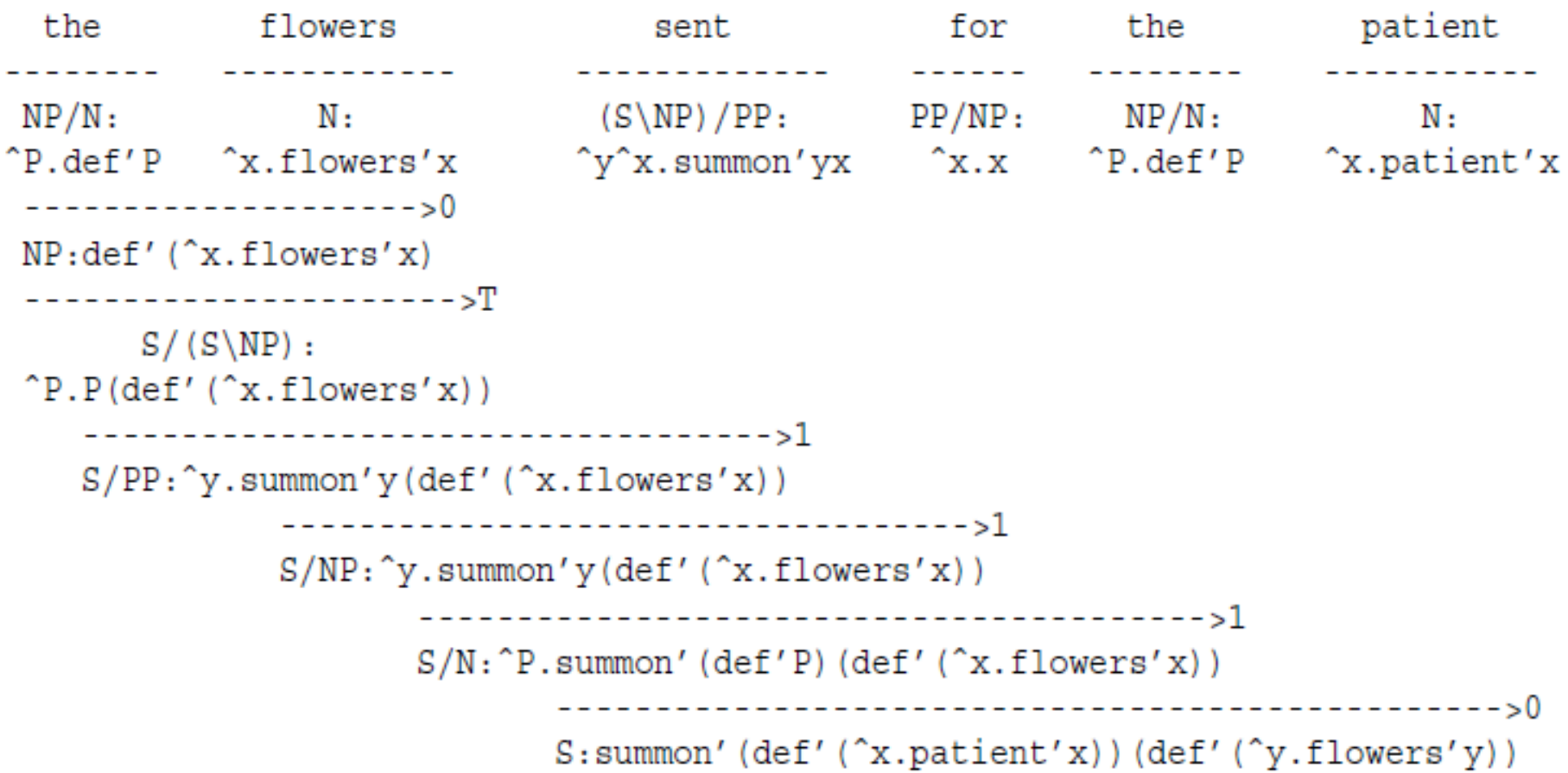


Figure: Incremental CCG derivation (Figure taken from McConville's PhD thesis.)

So how does that work?

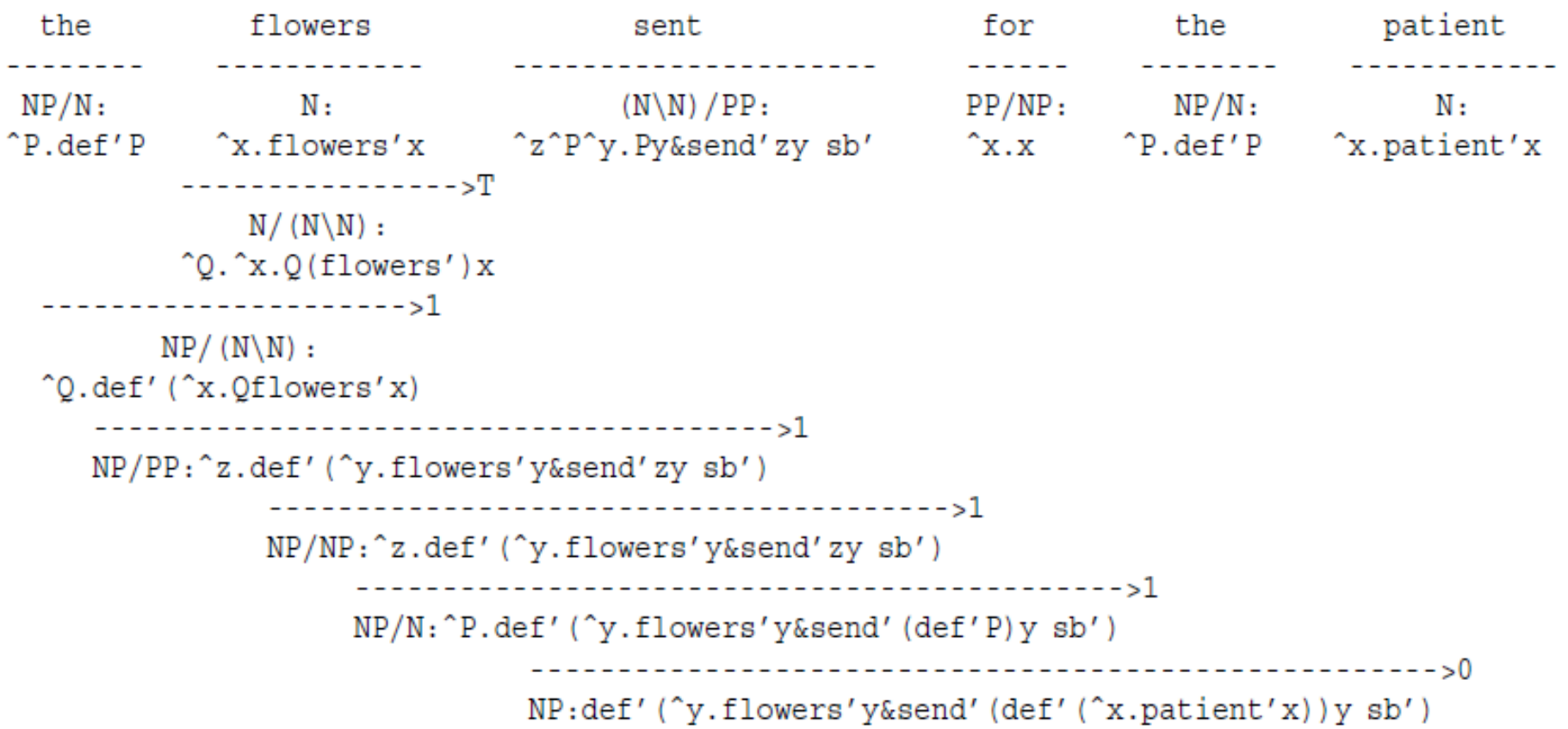


Figure: Incremental CCG derivation (Figure taken from McConville's PhD thesis.)

Table of Contents

Incrementality in CCG

But is an incremental derivation *always* possible?

In particular, let's look at our psycholinguistic data and see whether CCG can explain those.

Incrementality in CCG

But is an incremental derivation *always* possible?

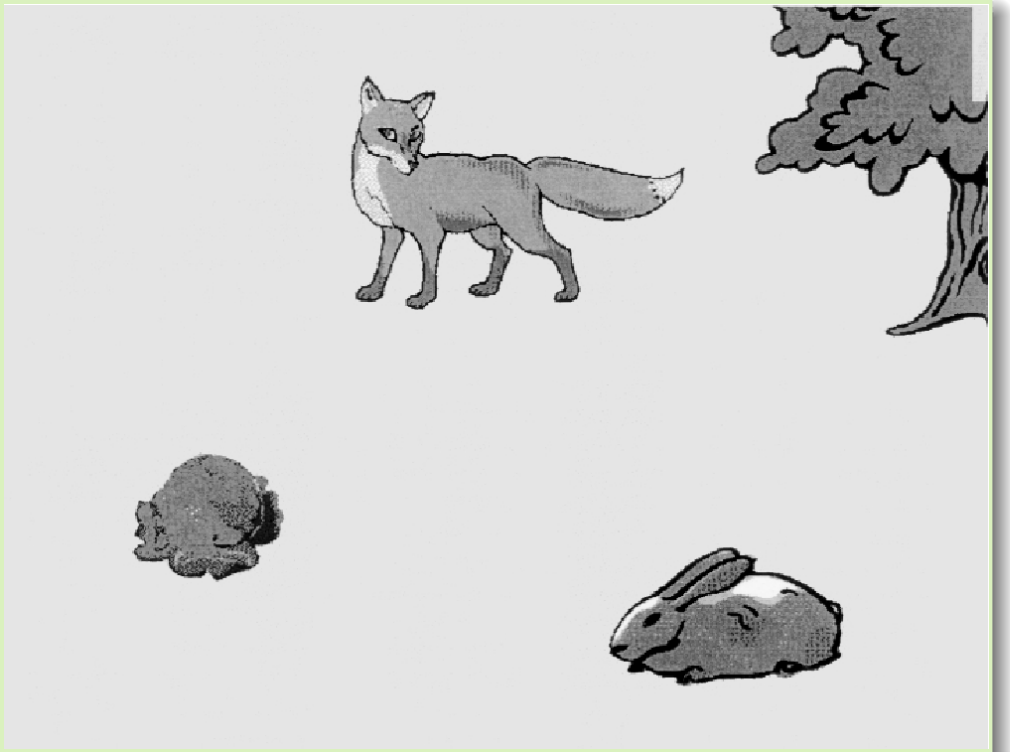
In particular, let's look at our psycholinguistic data and see whether CCG can explain those.

"Der Hase frisst gleich den Kohl."

The Hare-nom will eat soon the cabbage-acc.

"Den Hasen frisst gleich der Fuchs."

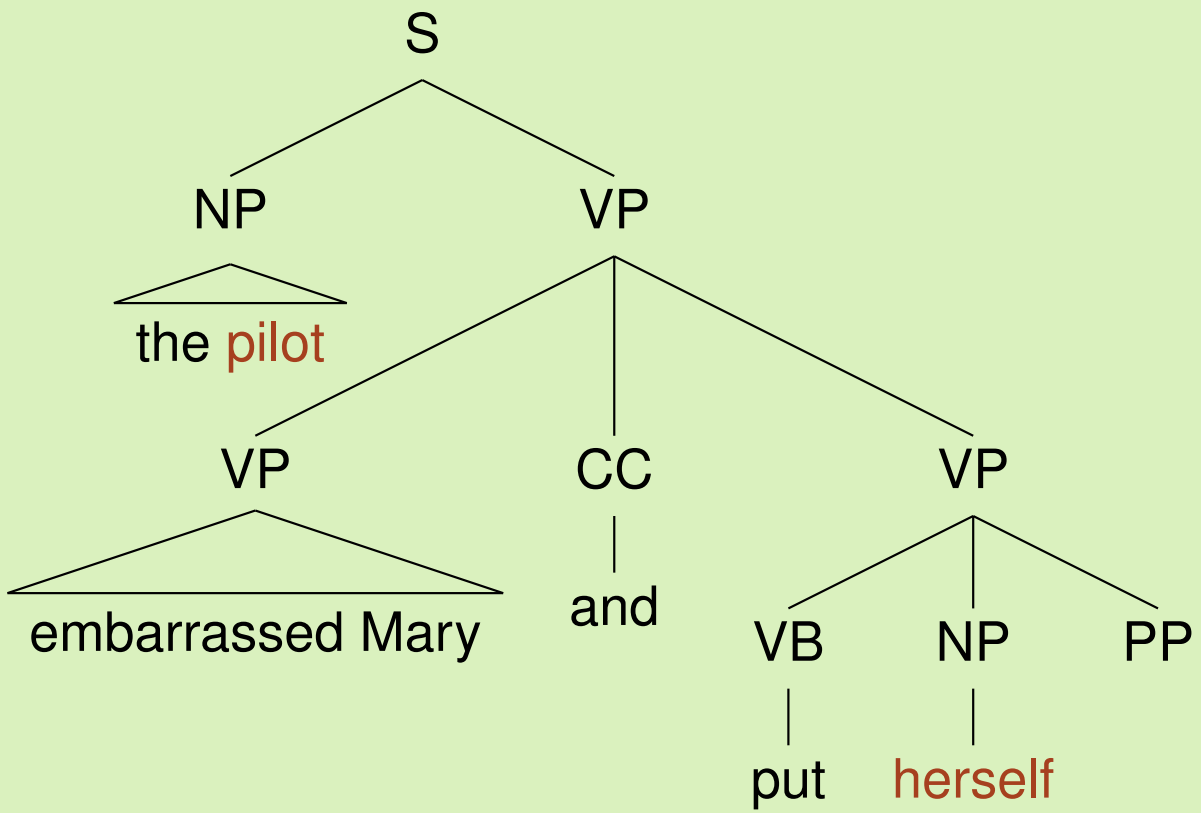
The Hare-acc will eat soon the fox-nom.



Incrementality in CCG

But is an incremental derivation *always* possible?

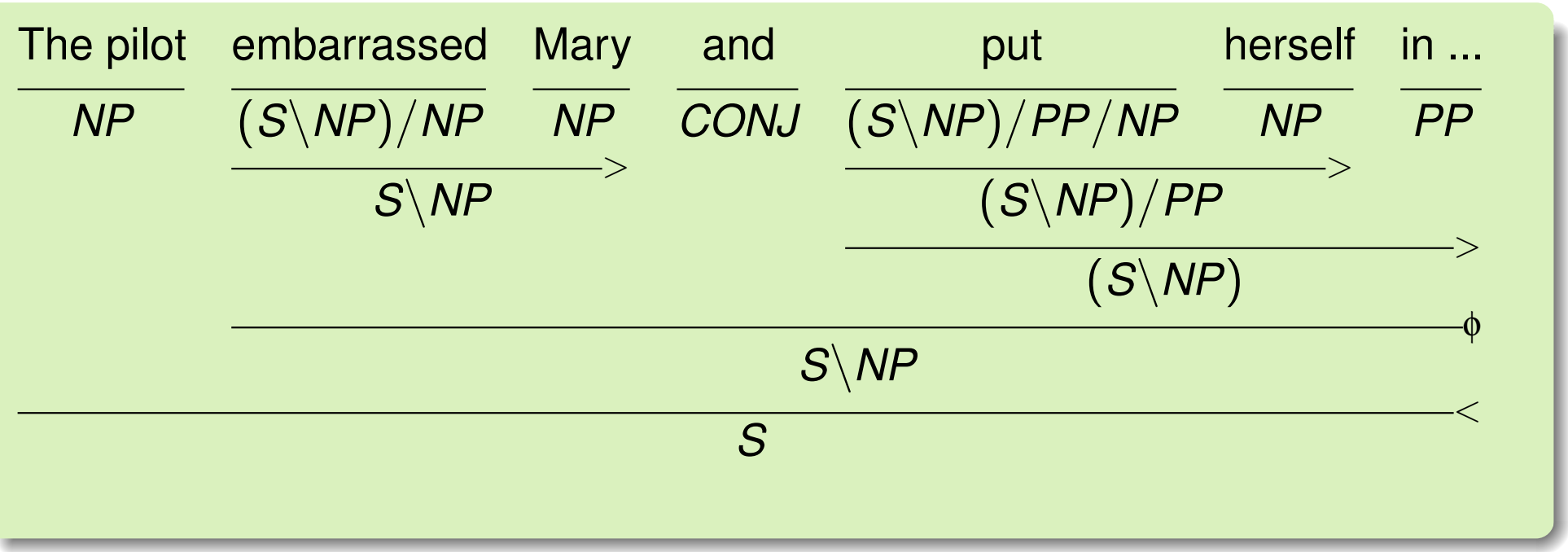
In particular, let's look at our psycholinguistic data and see whether CCG can explain those.



Incrementality in CCG

But is an incremental derivation *always* possible?

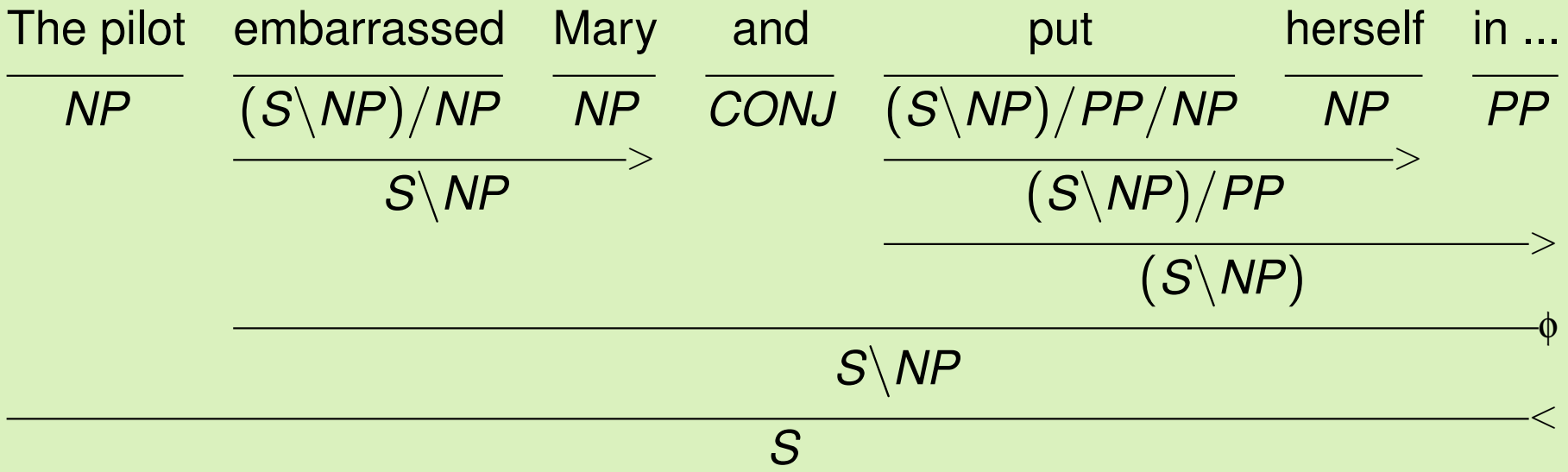
In particular, let's look at our psycholinguistic data and see whether CCG can explain those.



Incrementality in CCG

But is an incremental derivation *always* possible?

In particular, let's look at our psycholinguistic data and see whether CCG can explain those.

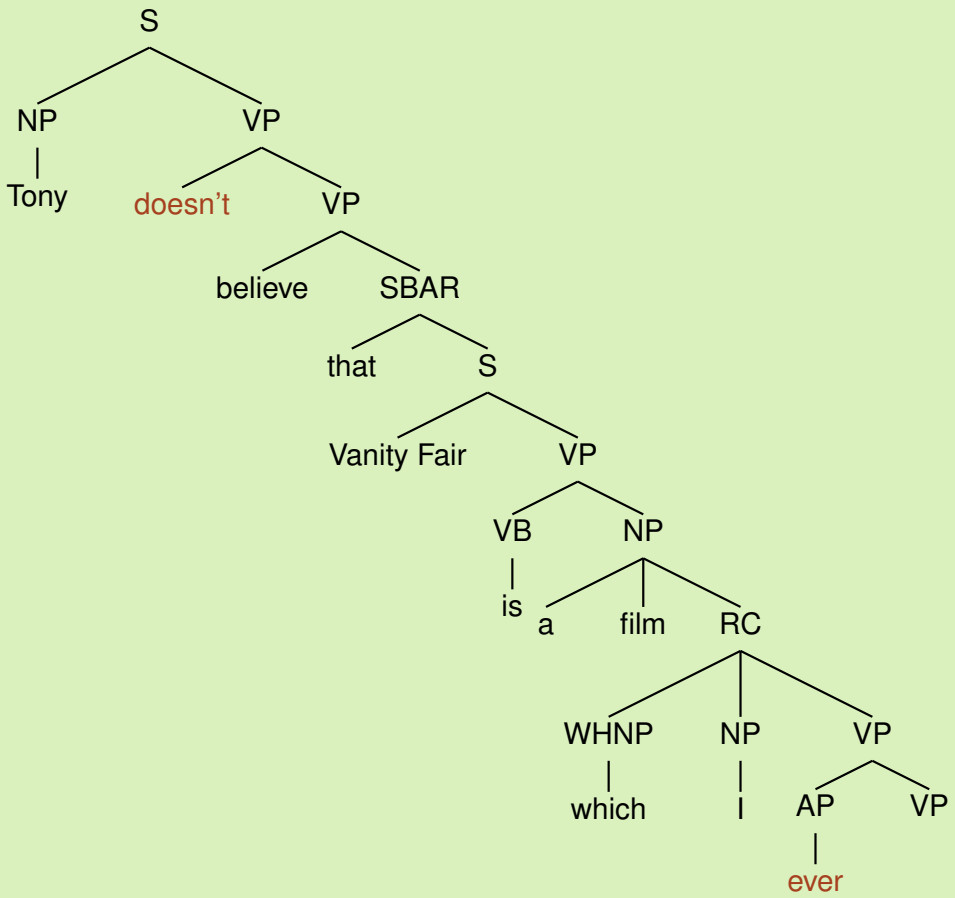


Can't derive incrementally because coordination requires identical categories.

Incrementality in CCG

But is an incremental derivation *always* possible?

In particular, let's look at our psycholinguistic data and see whether CCG can explain those.



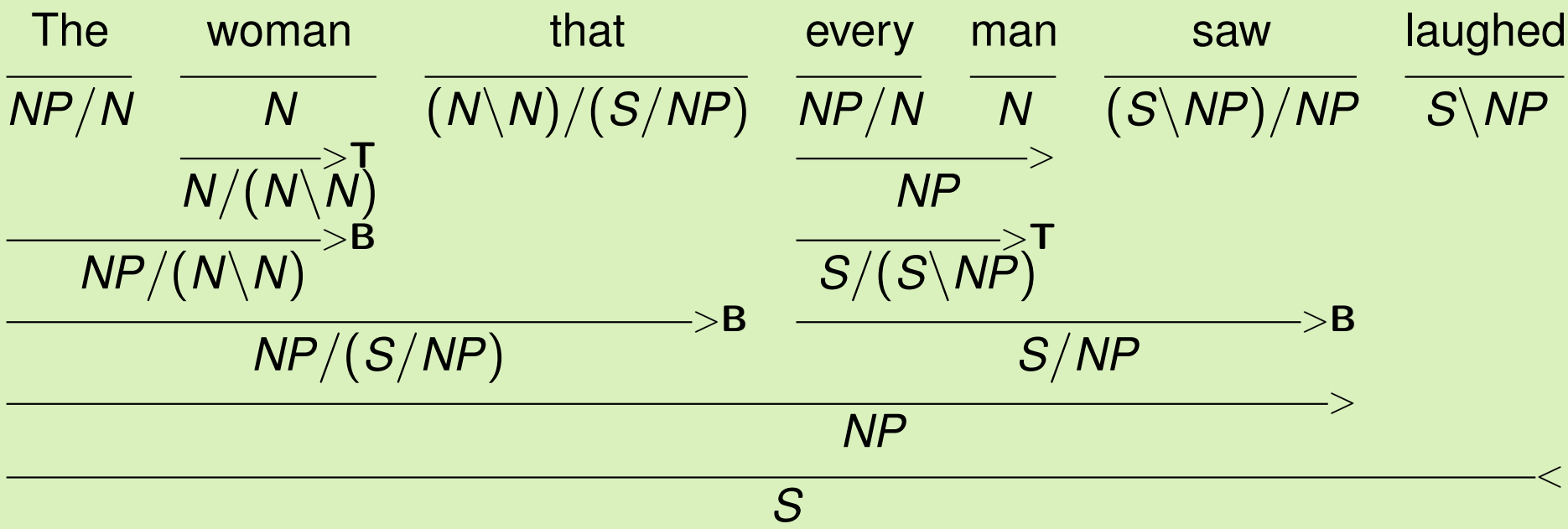
Let's break this down and try whether we can derive object relative clauses incrementally.

Incrementality in CCG

But is an incremental derivation *always* possible?

In particular, let's look at our psycholinguistic data and see whether CCG can explain those.

Most incremental standard CCG derivation:



Incrementality in CCG

But is an incremental derivation *always* possible?

In particular, let's look at our psycholinguistic data and see whether CCG can explain those.

Most incremental derivation with Geach rule:

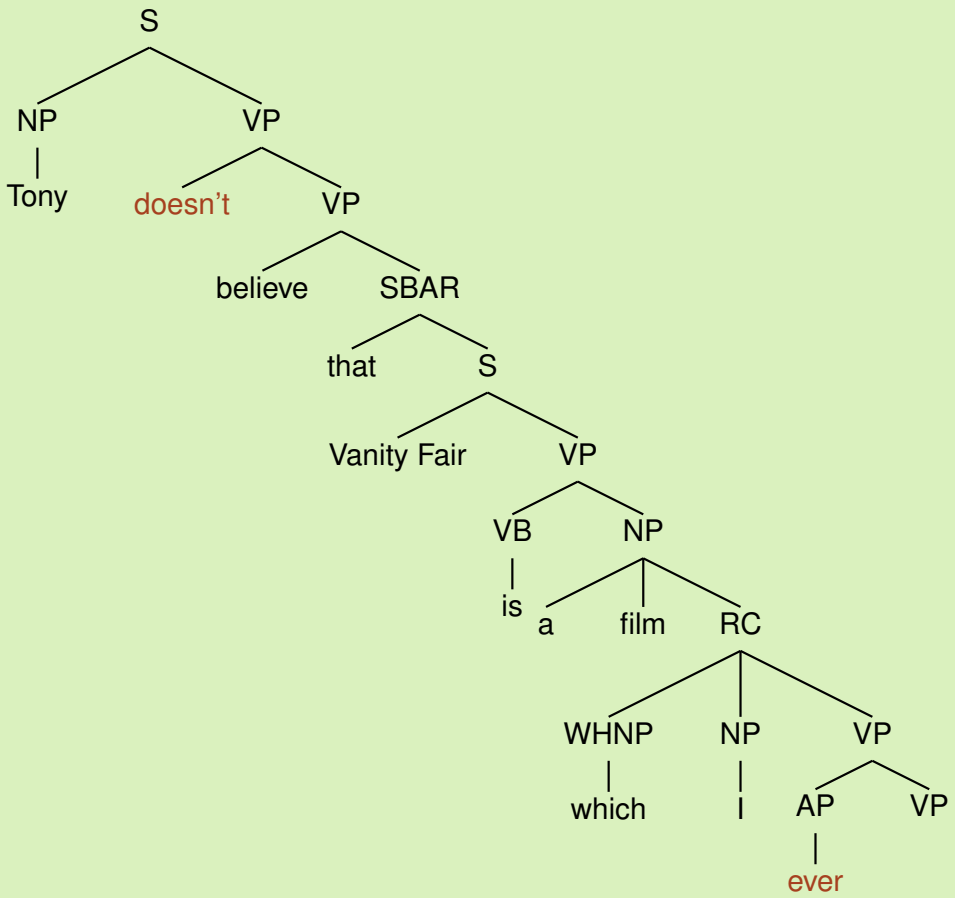
$Y/Z \quad \Rightarrow_{\mathbf{B}} \quad (Y/G)/(Z/G)$ (normally wrapped in Composition rules)

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{woman}}{N}}{N/(N\backslash N)}>\mathbf{T} \quad \dfrac{\text{that}}{(N\backslash N)/(S/NP)}}{N/(S/NP)}>\mathbf{B} \quad \dfrac{\dfrac{\dfrac{\text{every}}{NP/N} \quad \dfrac{\text{man}}{N}}{NP}> \quad \dfrac{\text{saw}}{(S\backslash NP)/NP}}{S/(S\backslash NP)}>\mathbf{T}}{(S/NP)/((S\backslash NP)/NP)}\mathbf{B}}{N/((S\backslash NP)/NP)}>\mathbf{B}}{N}>$$

Incrementality in CCG

But is an incremental derivation *always* possible?

In particular, let's look at our psycholinguistic data and see whether CCG can explain those.



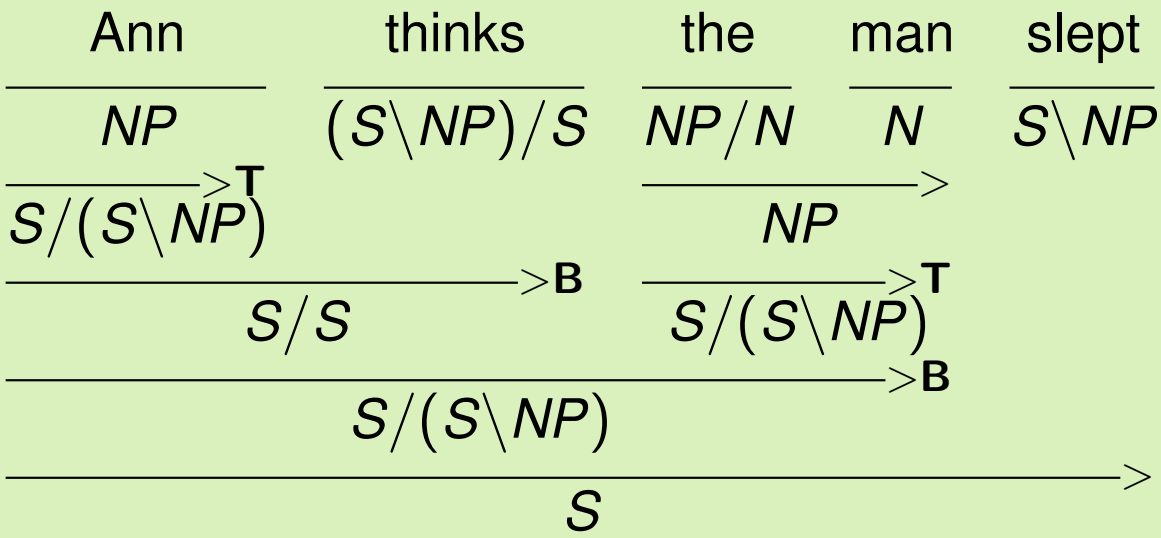
Can do this example, if the use of unary Geach rule is allowed.

Incrementality in CCG

But is an incremental derivation *always* possible?

In particular, let's look at our psycholinguistic data and see whether CCG can explain those.

More cases: Connected derivation of complement clause not possible



Conclusions on CCG and Incrementality

- Common constructions including object relative clauses, complement clauses and some cases of coordination do not allow for incremental derivation in CCG.
- Affected prefixes not a CCG constituent, hence assumed not to have a semantic interpretation.
- Alternatives that would allow for incremental CCG derivations:
  - new categories for certain words ($\rightarrow$ overgeneration)
    would accept *[the man that every] and [the woman that no] kid saw slept*, which violates island constraint
  - additional combinatory rules ($\rightarrow$ overgeneration)
  - top-down or left-corner parsing as opposed to bottom-up ($\rightarrow$ tractability?)