university of groningen

Language and Inference

Day 3: Building Meaning Representations

Johan Bos
johan.bos@rug.nl

Today

- Introduce a method to build meaning representations from English text
- Introduce a grammar formalism
- Specify the syntax-semantics interface

Combinatory Categorial Grammar

CCG is a lexicalised theory of grammar

- Many different lexical categories
- Few grammar rules (based on combinatory logic)
- Covers complex cases of coordination and long-distance dependencies

Not just theory, also used in practice!

- OpenCCG (Baldridge, White)
- CCGbank (Hockenmaier)
- Groningen Meaning Bank
- C&C supertagger and parser (Clark, Curran)

Combinatory Categorial Grammar

Basic Categories

S	sentence
NP	noun phrase
N	noun
PP	prepositional phrase

The category S comes with a feature to distinguish between various sentence mood and verb phrase forms

NP/N	determiner
N/N	adjective
$S_{dcl}\backslash NP$	verb phrase (declarative mood)
$(S_{ng}\backslash NP)/NP$	transitive verb (present perfect)
$(S_X\backslash NP)\backslash(S_X\backslash NP)$	adverb
$(N\backslash N)/NP$	preposition

The direction of the slash determines where the argument appears: forward slash (/): right; backward slash (\): left

Functor Categories

Word		Category
boy	:	N
everything	:	NP
the	:	NP/N
eats	:	$S_{dcl}\backslash NP$
eats	:	$(S_{dcl}\backslash NP)/NP$
quickly	:	$(S_X\backslash NP)\backslash(S_X\backslash NP)$

Example Lexicon

Combinatory Rules of CCG

Application
- Forward **>**
- Backward **<**

Composition
- (Generalised) Forward **>B**
- (Generalised) Backward **<B**

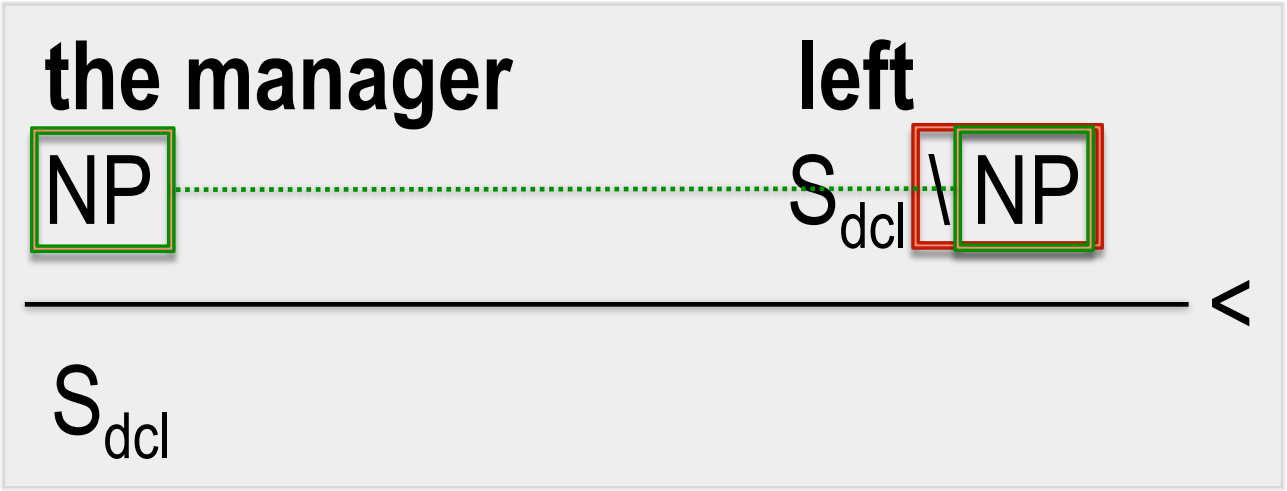
Crossed Composition
- (Generalised) Forward **>Bx**
- (Generalised) Backward **<Bx**

Type Raising
- Forward **>T**
- Backward **<T**

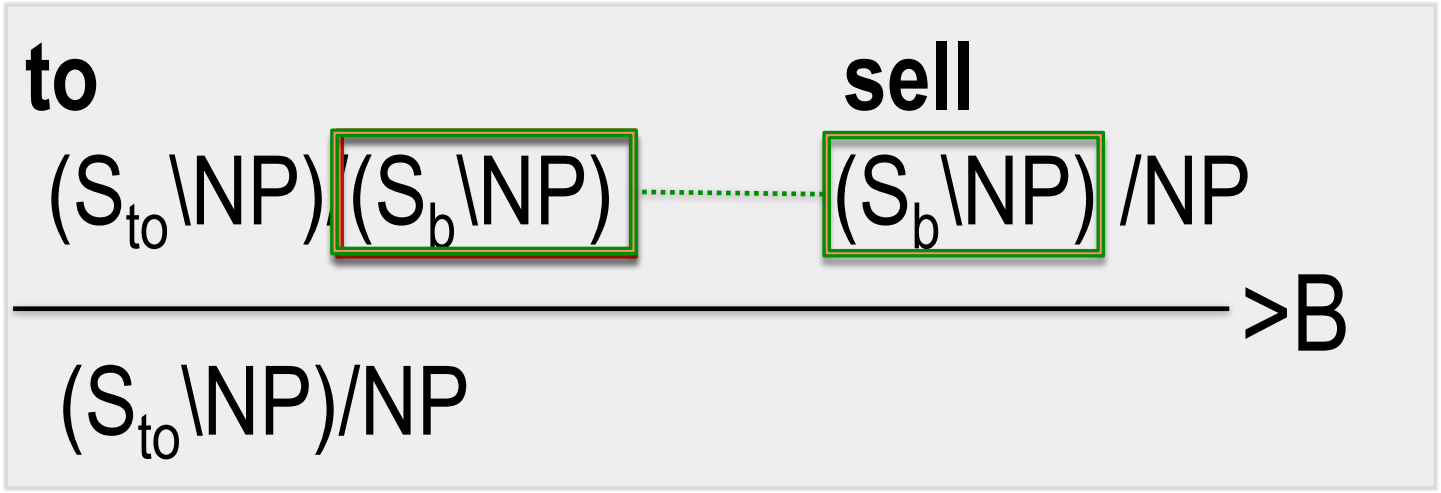
Substitution
- Forward **>S**
- Backward **<S**

Crossed Substitution
- Forward **>Sx**
- Backward **<Sx**

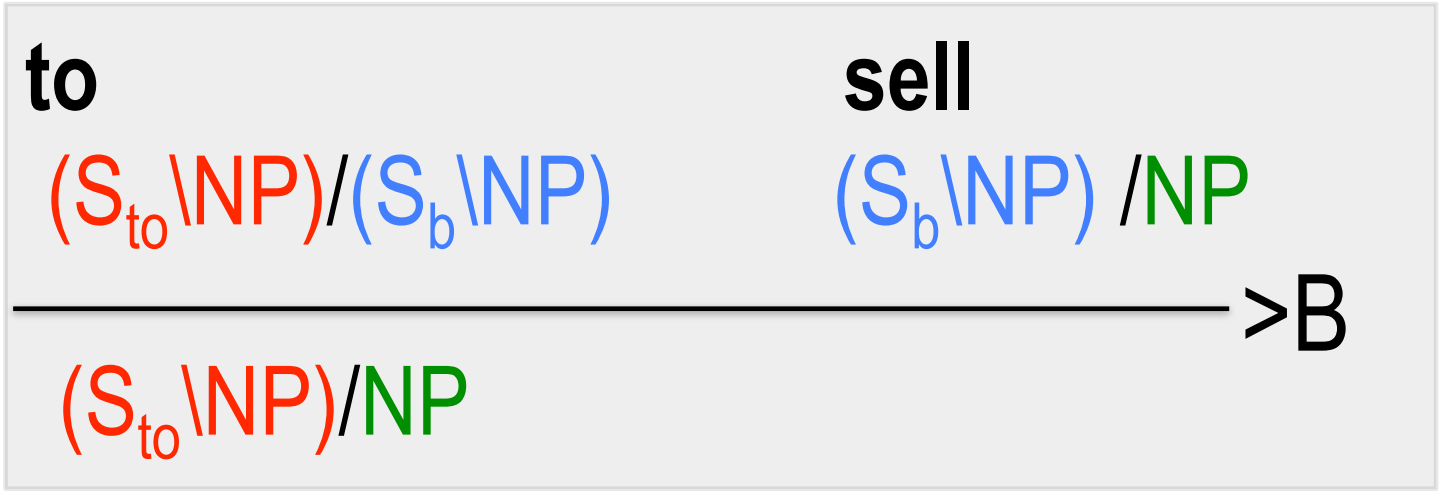
Forward Application (>)



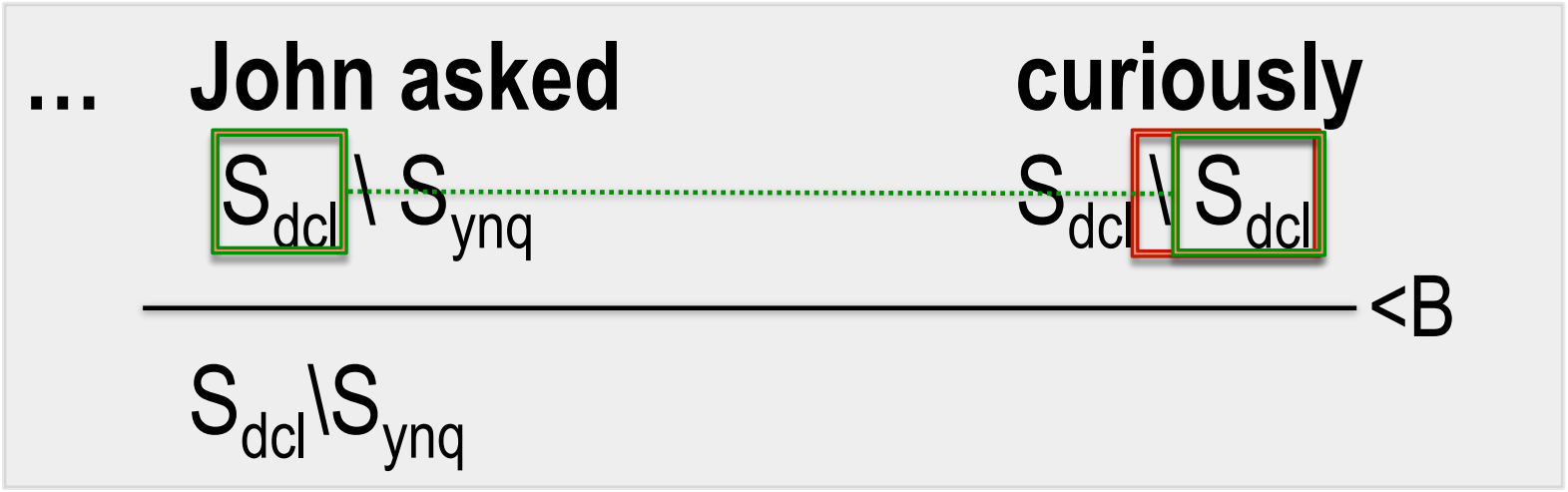
Backward Application (<)



Forward Composition (>B)

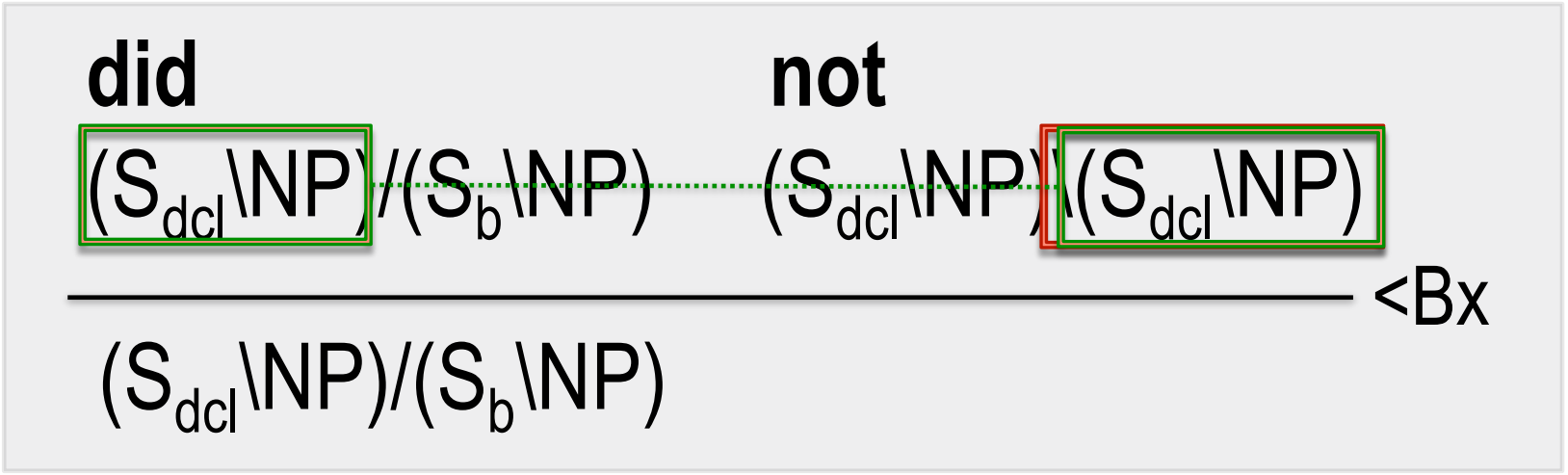


Forward Composition (>B)

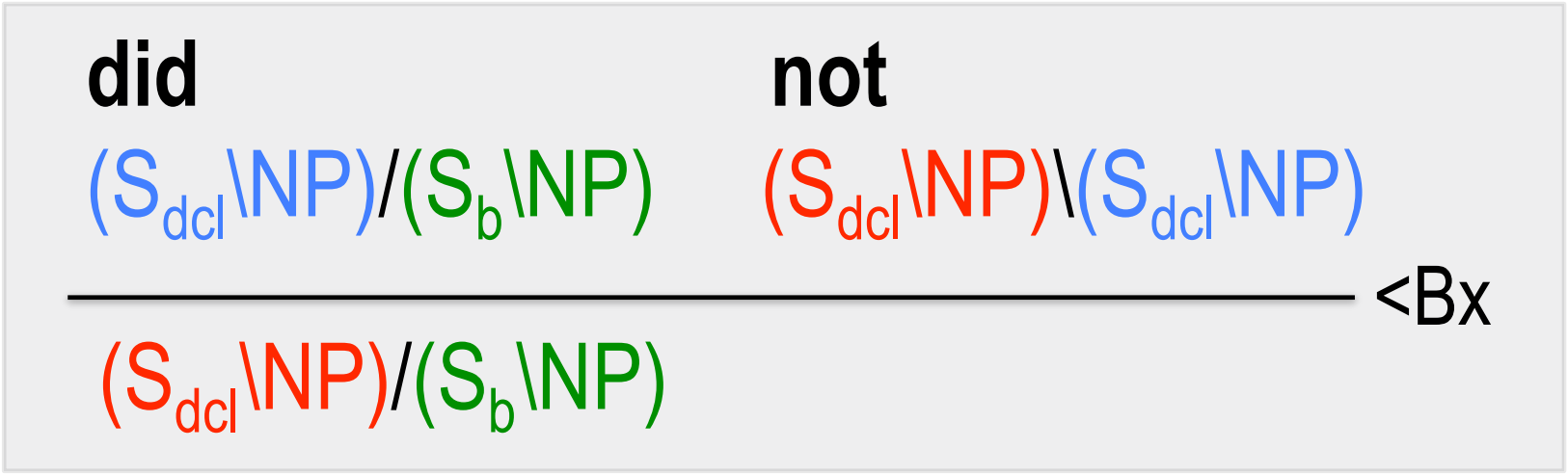


Backward Composition (<B)

Backward Composition (<B)



Backward Crossed Composition (<Bx)



Backward Crossed Composition (<Bx)

$$\frac{X/Y \quad Y}{X} >$$

$$\frac{Y \quad X\backslash Y}{X} <$$

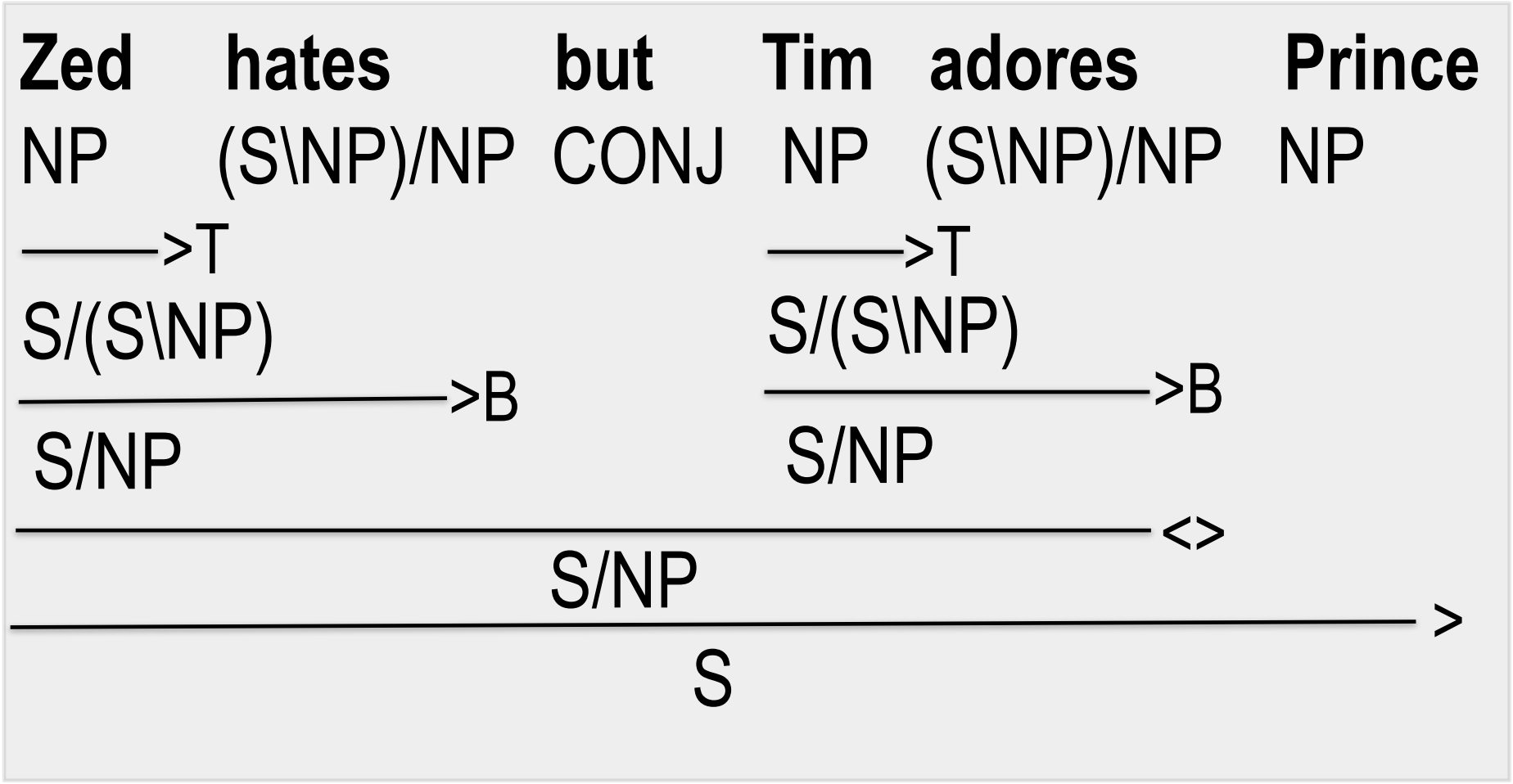
$$\frac{X/Y \quad Y/Z}{X/Z} >B$$

$$\frac{Y\backslash Z \quad X\backslash Y}{X\backslash Z} <B$$

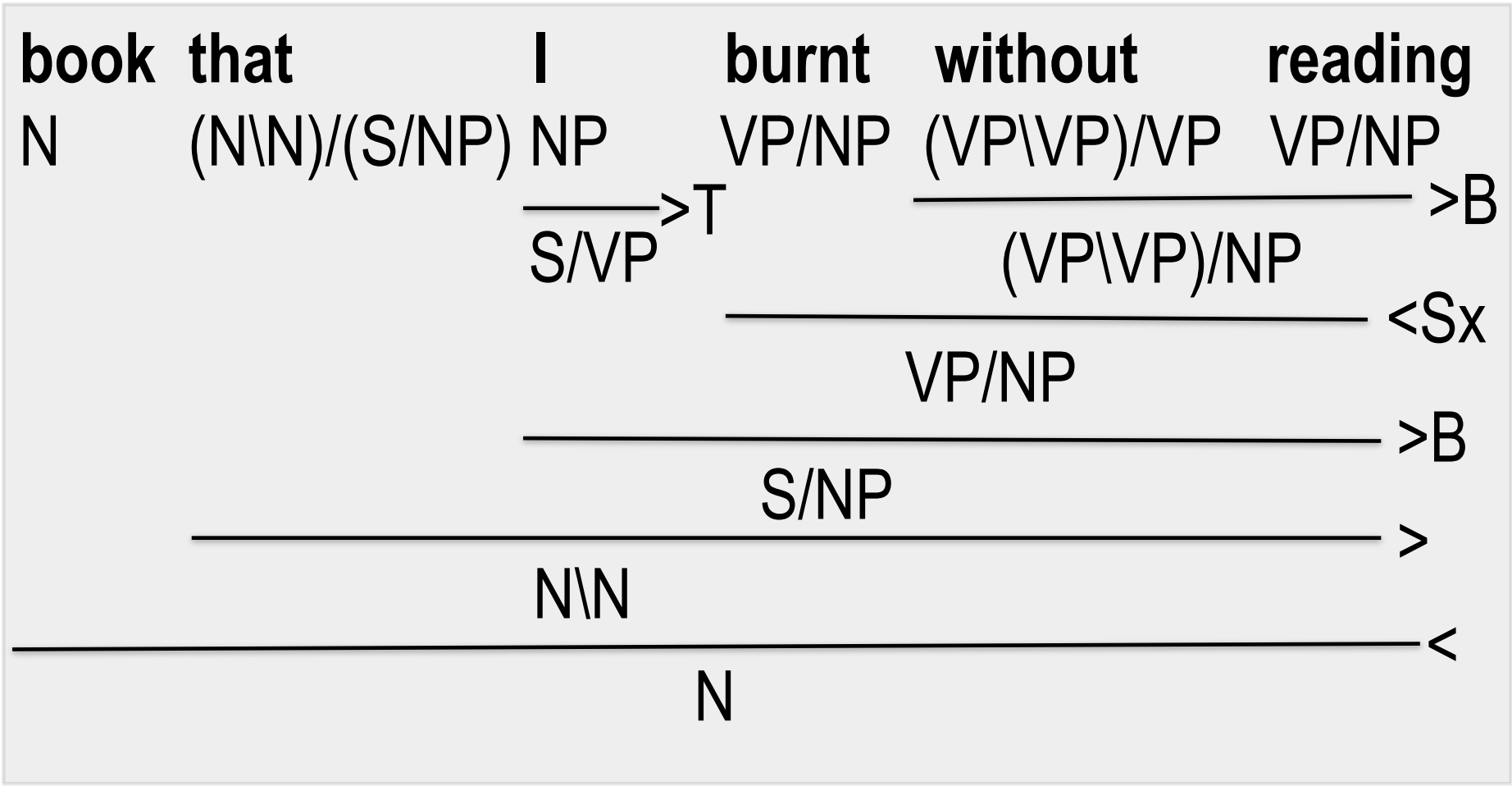
$$\frac{X/Y \quad Y\backslash Z}{X\backslash Z} >Bx$$

$$\frac{Y/Z \quad X\backslash Y}{X/Z} <Bx$$

Rule schemata (1)



Type Raising (>T) and Coordination (<>)



Substitution (S), “parasitic gap”

$$\frac{X}{Y/(Y\backslash X)}\ >T \qquad \frac{X}{X\backslash(Y/X)}\ <T$$

$$\frac{X \quad CONJ \quad X}{X}\ <\!>$$

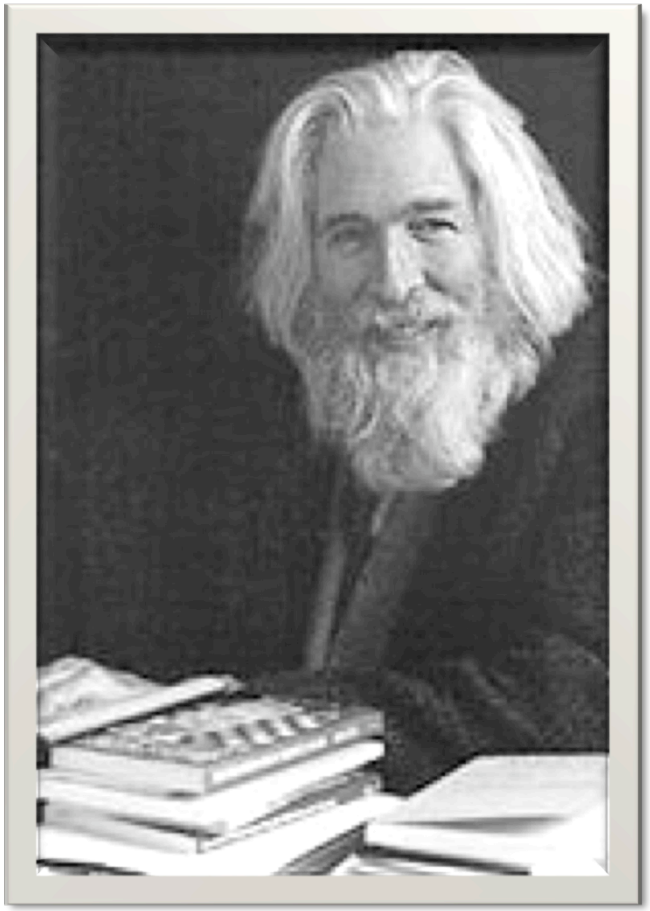
$$\frac{(X/Y)/Z \quad Y/Z}{X/Z}\ >S \qquad \frac{Y/Z \quad (X\backslash Y)/Z}{X/Z}\ <Sx$$

Rule schemata (2)

Bluebird

Starling

Thrush



Raymond Smullyan

The C&C parser

The C&C parser consists of two steps:

1. Supertagging
   (= assigning lexical categories to tokens)

2. Parsing
   (= building CCG derivations)

C&C supertagger and parser

- Implementation of fast supertagger and parser trained on Hockenmaier's CCGbank
- Supertagger assigns CCG categories to words
- Integrated with POS tagging and named entity recognition
- Reaching high speed
  (a minute to parse one section of the WSJ)

Morphology

- The C&C tools also use the morphological analyser *Morpha*
- Morpha returns the lemma and inflection type of a word, given the word form and the POS

EXAMPLE

see (VB)	=> see
seen (VBN)	=> see
saw (VBD)	=> see
saw (NN)	=> saw
saws (NNS)	=> saw

Hockenmaier's CCGbank

- A large collection of CCG trees
  - 49,000 sentences, over a million of tagged words
  - A total of 1,286 different categories (of which 847 appear more than once)
- Extracted from the Penn Treebank
  - *Wall Street Journal* newspaper texts
  - All 25 sections