Meaning, Form and Adjacency: Schönfinkel's Legacy

Cem Bozşahin

Computer Engineering & Cognitive Science
Middle East Technical University (METU), Ankara
bozsahin@metu.edu.tr

Ankara Linguistic Circle, October 10, 2008

Moses Ilyich Schönfinkel

1889 Dnipropetrovsk, Ukraine – 1942 Moscow

December 7, 1920, Göttingen seminar to David Hilbert's research group

$$f(x) = x + 1 \qquad f(3) = 3 + 1$$

$$\lambda x.x + 1 \qquad \lambda x.x + 1(3) =_\beta 3 + 1$$

$$f(x,y) = x - y \qquad f(3,5) = 3 - 5$$

$$\lambda x \lambda y.x - y \qquad \lambda x \lambda y.x - y(3) =_\beta \lambda y.3 - y$$

$$\lambda y \lambda x.x - y \qquad \lambda y \lambda x.x - y(5) =_\beta \lambda x.x - 5$$

Two definitions of $f(x,y)$ must be related

Polish notation. Functions first: $f(x,y) = x - y = -xy$

$-xy = -yx$ is clearly wrong

$-xy = \mathbf{C} - yx$ is fine

if $\mathbf{C}fbc = fcb$

C is Schönfinkel's **interchange**, T (nowadays called **permutator**)

Relating functions in function-first notation: $f(gx)$

$f(gx)$ depends on x because g depends on x.

"*That John sleeps worries* ..." function $worry'(sleep'john')$

$\mathbf{B}fga = f(ga)$

B is Schönfinkel's **compositor** (he called it Z)

These **combinators** get their arguments one at a time: $\mathbf{B}fga = ((\mathbf{B}f)g)a$

$\mathbf{B}worry'sleep'john' = worry'(sleep'john')$

B, **C** are invariants; they do the same thing to any function

Schönfinkel (1924) showed that only a handful of combinators can represent any conceivable relation between functions and arguments.

I $f = f$

K $fa = f$ **T** $af = fa$

S $fga = fa(ga)$ (called fusion by MIS)

D $fgha = fg(ha)$

Φ $fgha = f(ga)(ha)$

Definitive work on combinators: Curry and Feys (1958)

All combinators have two things in common:

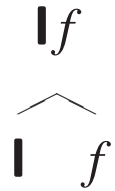
They get their arguments one-at-a-time

They apply the same operation blindly

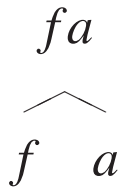
They arise from a **single semantic base**: juxtaposition of f and a as fa

Any function-argument relation is binary: $fabcd = ((((fa)b)c)d)$

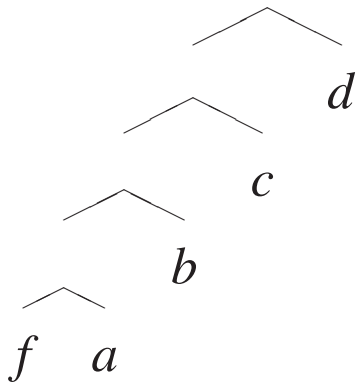
Including zero-argument functions: $f = \mathbf{I}f$



and one-argument functions: fa



Currying: from $f(a,b,c,d)$ to $((((fa)b)c)d) = fabcd$:



Schönfinkeling alive in secret sects.

Haskell Brooks Curry 1900 – 1982

Syntactic Reflex of Combinators

Semantic primitive **juxtaposition** (fa) needs a functor and an argument:

$$\mathbf{X/Y}\text{: } f \quad \mathbf{Y}\text{: } a \quad \rightarrow \quad \mathbf{X}\text{: } fa \qquad (>)$$

A non-functor such as ba is not possible:

$$\mathbf{X}\text{: } b \quad \mathbf{Y}\text{: } a \quad \rightarrow \quad \mathbf{X}\text{: } ba \qquad (*>)$$

Semantics says nothing about syntactic order of f and a:

$$\mathbf{Y}\colon a \quad \mathbf{X}\backslash\mathbf{Y}\colon f \quad \rightarrow \quad \mathbf{X}\colon fa \qquad (<)$$

Semantics does say something about which one is syntactically functor:

$$\mathbf{Y}\colon a \quad \mathbf{X}\backslash\mathbf{Y}\colon f \quad \rightarrow \quad \mathbf{X}\colon af \qquad (^{*}<)$$

$$\mathbf{Y}\colon f \quad \mathbf{X}\backslash\mathbf{Y}\colon a \quad \rightarrow \quad \mathbf{X}\colon fa \qquad (^{*}<)$$

Semantics says composing functions require syntactic functors:

$$\mathbf{X/Y} : f \quad \mathbf{Y/Z} : g \quad \mathbf{Z} : a \rightarrow \quad \mathbf{X/Z} : f(ga) \qquad (>\mathbf{B})$$

$$\mathbf{X/Y} : f \quad \mathbf{Y} : g \quad \mathbf{Z} : a \rightarrow \quad \mathbf{X/Z} : f(ga) \qquad (^{*}>\mathbf{B})$$

Semantics does say non-binary combinations are spurious:

$$\mathbf{X/Y} : f \quad \mathbf{Y/Z} : g \quad \rightarrow \quad \mathbf{X/Z} : \lambda x.f(gx) \qquad (>\mathbf{B})$$

Semantics says that syntactic types cannot be arbitrary:

$$\mathbf{X/Y} : f \quad \mathbf{Y/Z} : g \quad \rightarrow \quad \mathbf{X/W} : \lambda x.f(gx) \qquad (^*>\mathbf{B})$$

Semantics cannot say anything about syntactic order of f and g:

$$\mathbf{Y\backslash Z} : g \quad \mathbf{X\backslash Y} : f \quad \rightarrow \quad \mathbf{X\backslash Z} : \lambda x.f(gx) \qquad (<\mathbf{B})$$

Syntax says some types are subsumed by others:

$$\textbf{X/(X}\backslash\textbf{Z)}\text{: }\lambda g.ga \quad \textbf{X}\backslash\textbf{Z}\text{: } f \quad \rightarrow \quad \textbf{X}\text{: } fa \qquad (^{*}{>}\ \textbf{T})$$

But not all:

$$\textbf{Z}\text{: } a \quad \rightarrow \quad \textbf{X/(X}\backslash\textbf{Z)}\text{: }\lambda f.fa \qquad ({>}\ \textbf{T})$$

Consistent directionality of types acts as a projection principle:

$$\mathbf{X/Y} : f \quad \mathbf{Y/Z} : g \quad \rightarrow \quad \mathbf{X/Z} : \lambda x.f(gx) \qquad (>\mathbf{B})$$

$$\mathbf{X/Y} : f \quad \mathbf{Y/Z} : g \quad \rightarrow \quad \mathbf{X\backslash Z} : \lambda x.f(gx) \qquad (^{*}>\mathbf{B})$$

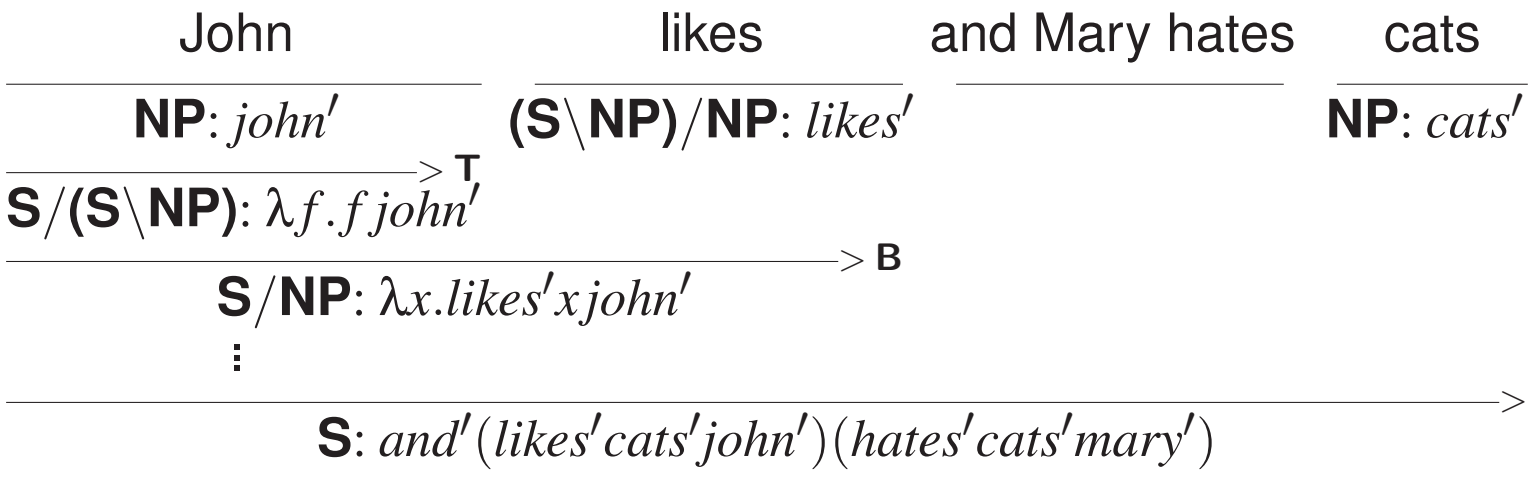
Having a unique source for semantics (adjacency) spells predictable syntactic machinery

Syntactic machinery does reflect semantic dependencies

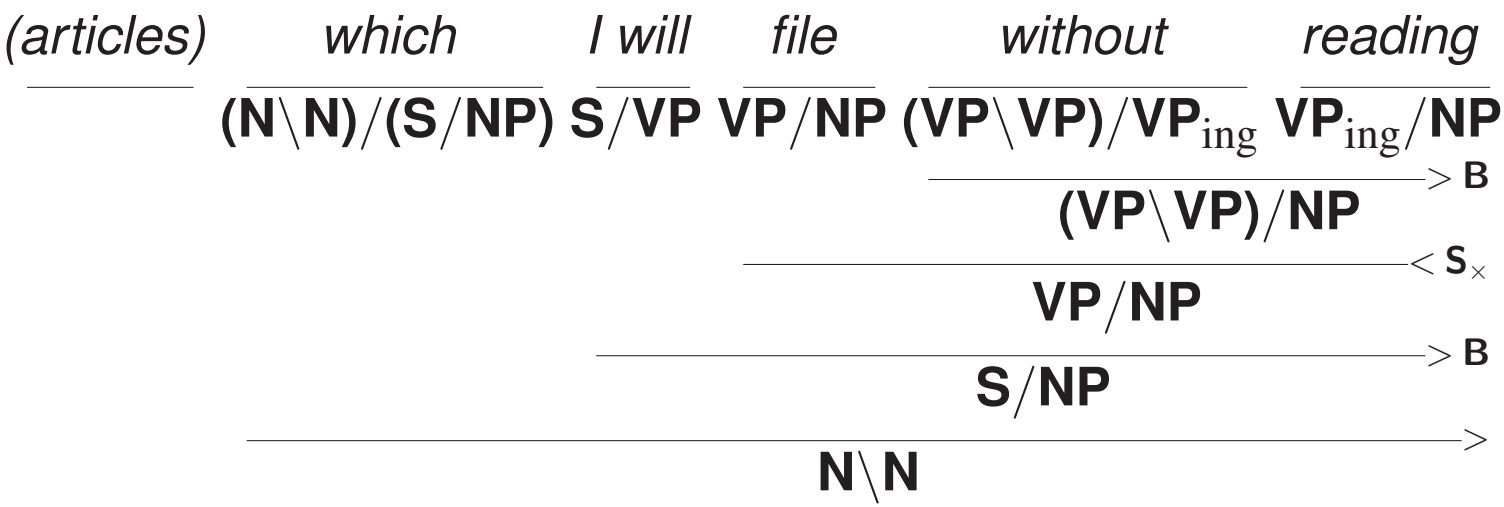
Once syntax is established combinatorially, compositional semantics comes for free

every step of the way.

Adjacency and displacement



Let **VP = S\NP**


(articles)
which
I will
file
without
reading
(N\N)/(S/NP)
S/VP
VP/NP
(VP\VP)/VP_ing
VP_ing/NP
>B
(VP\VP)/NP
<S×
VP/NP
>B
S/NP
>
N\N

(1)a. The cat that $[\text{John admires}]_{\mathbf{S/NP}}$ and $[\text{Mary hates}]_{\mathbf{S/NP}}$

b. *The cat that $[\text{John admires}]_{\mathbf{S/NP}}$ and $[\text{bites Mary}]_{\mathbf{S\backslash NP}}$

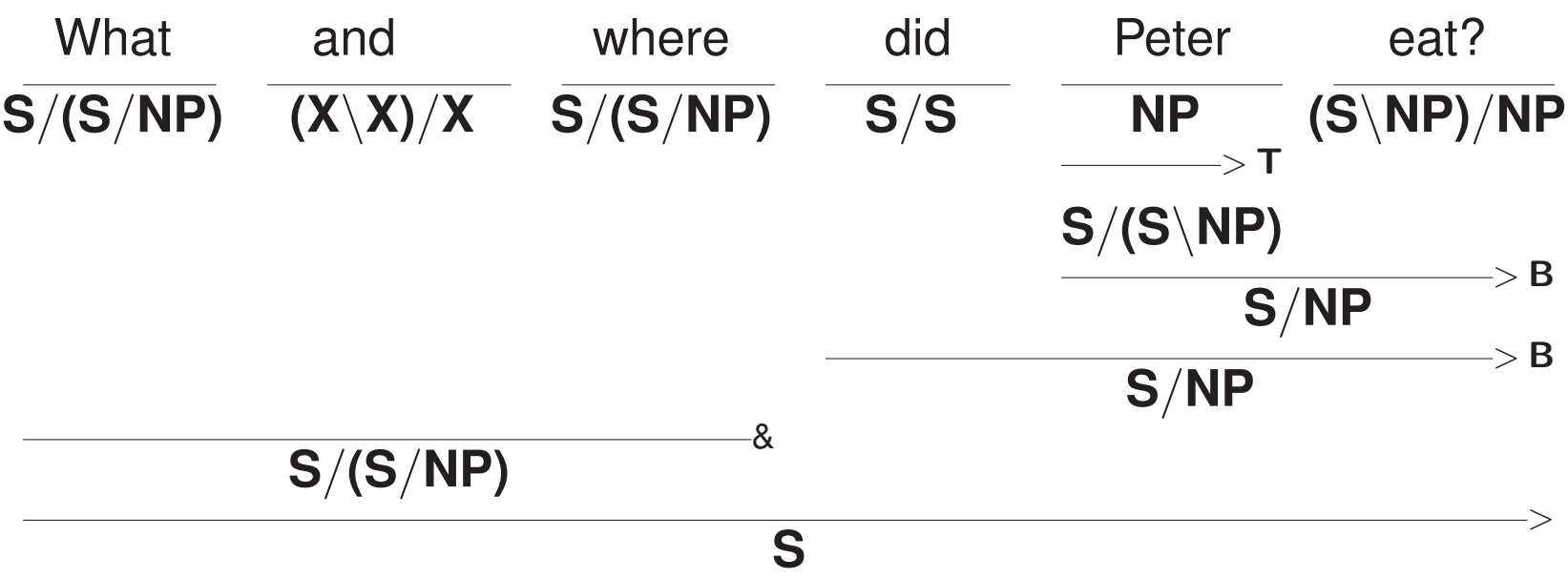
c. *The man that $[\text{admires John}]_{\mathbf{S\backslash NP}}$ and $[\text{Mary detests}]_{\mathbf{S/NP}}$

d. The man $[\text{that admires John}]_{\mathbf{N\backslash N}}$ and $[\text{(that) Mary detests}]_{\mathbf{N\backslash N}}$

e. *The cat that $[\text{John admires}]_{\mathbf{S/NP}}$ and $[\text{Mary hates it}]_{\mathbf{S}}$

f. *The cat that $[\text{John admires it}]_{\mathbf{S}}$ and $[\text{Mary hates}]_{\mathbf{S/NP}}$

Bi-clausal Multiple Wh-questions (BMW) Gracanin-Yüksek (2007)



Words versus phrases

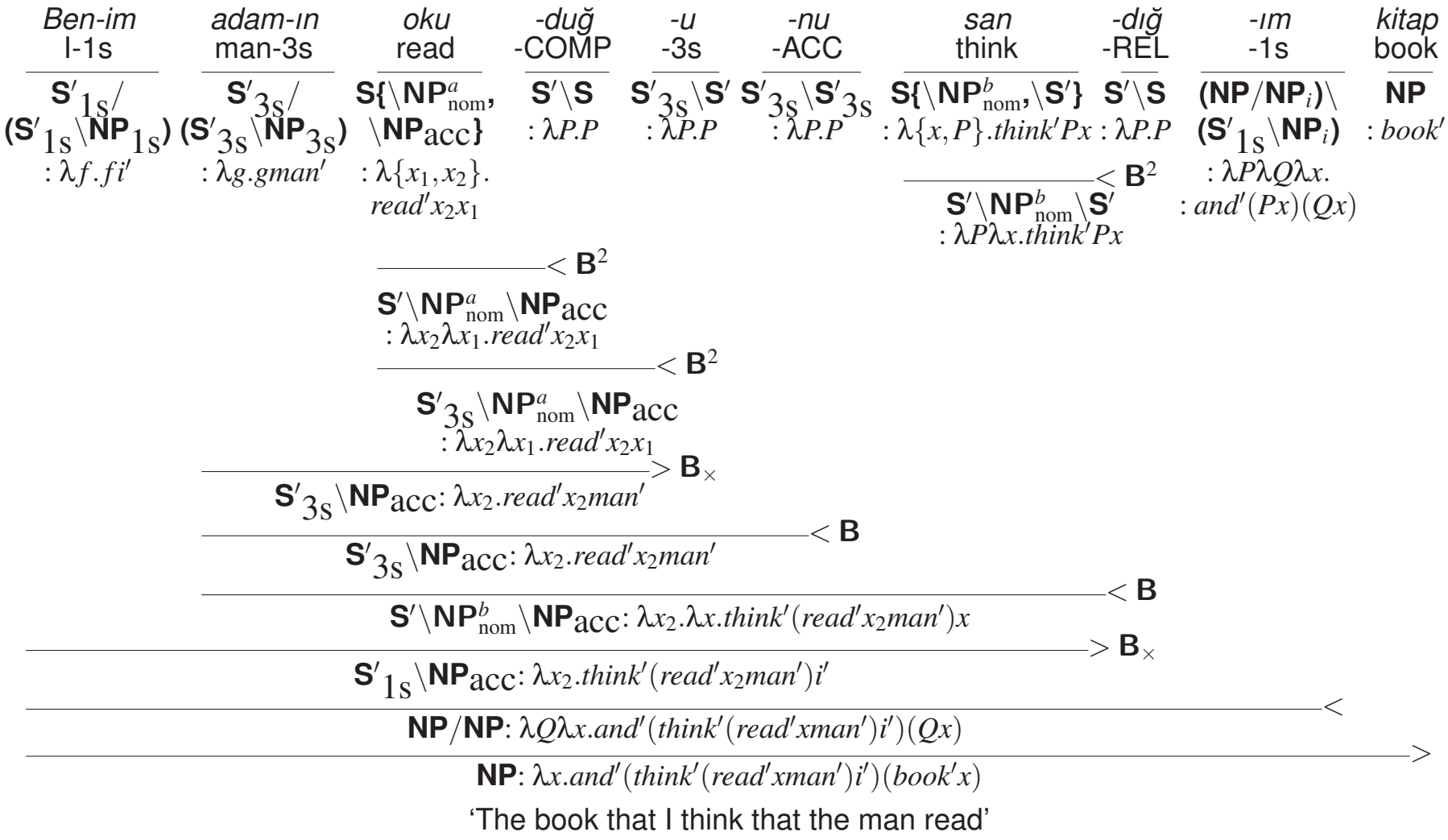
Word grammars

Phrase Grammars

Bridge Grammars

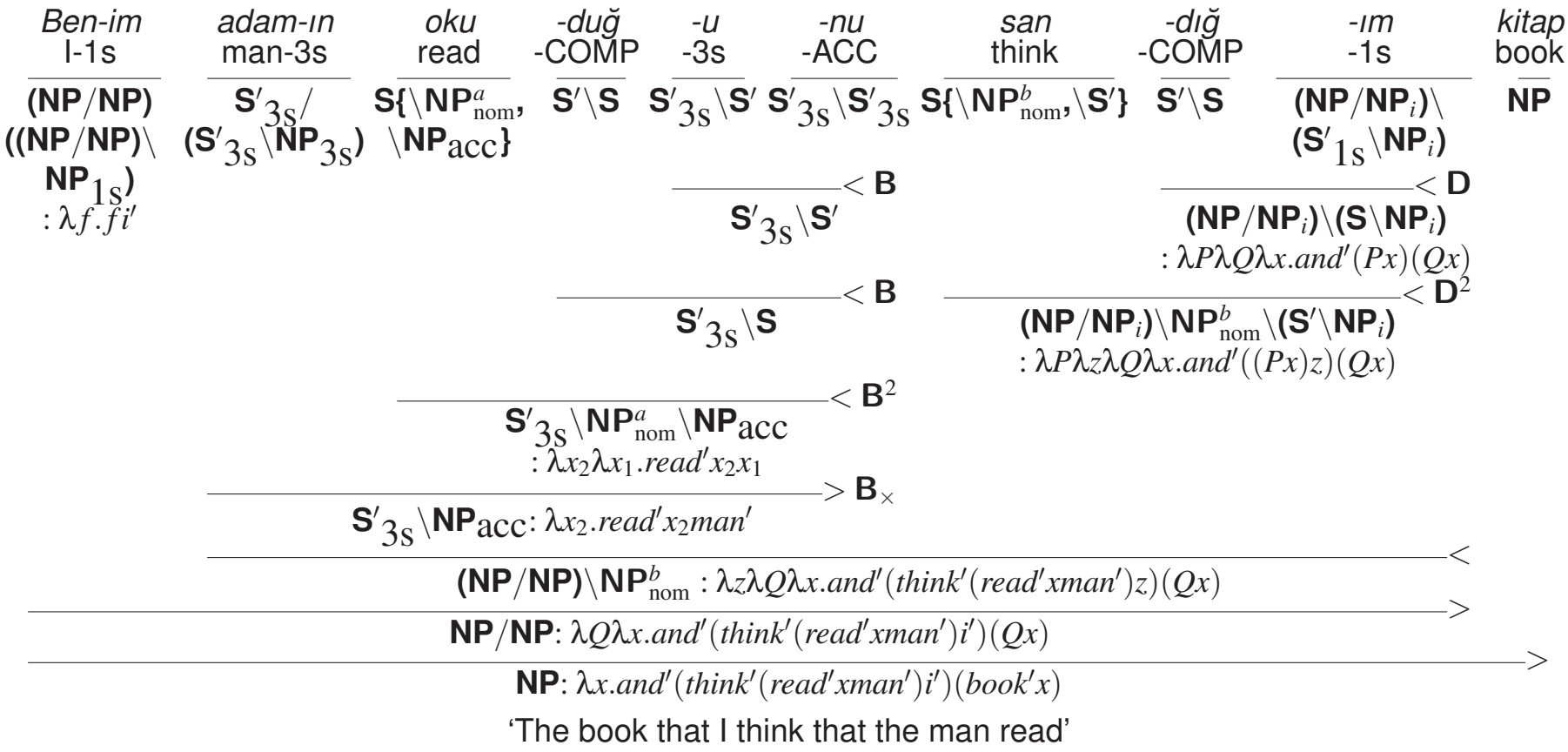
Lexical Integrity

Morphological versus Syntactic Mergers



'The book that I think that the man read'

No respect for word boundaries



'The book that I think that the man read'

All word boundaries are respected

Same lexical assumptions

Same derived semantics

Doing intra-word combinations first before inter-word combinations does not matter combinatorially.

Word/phrase distinction must be an interface problem.

Summary

Adjacency gave Schönfinkel his foothold into one-at-a-time interpretability of any function/argument configuration.

Polish (function-first) notation and elimination of variables complete the picture of primitive semantic operations as combinators par excellence.

When restricted to bounded use of **B**, **T**, **S**, we get mild context-sensitivity (Joshi, 1985; Vijay-Shanker and Weir, 1993),

Just like human languages.

Their syntactic counterparts are predictable,

due to adjacency and sequential channel.

1889 Dnipropetrovsk, Ukraine – 1942 Moscow

*References

Curry, Haskell B., and Robert Feys. 1958. *Combinatory Logic I*. Amsterdam: North-Holland.

Gracanin-Yüksek, Martina. 2007. "Coordinated Wh-Questions as Multidominance Structures." Cognitive Science Colloquium, METU, November 5.

Joshi, Aravind. 1985. "How Much Context-sensitivity is Necessary for Characterizing Structural Descriptions: Tree Adjoining Grammars." In David Dowty, Lauri Karttunen, and Arnold Zwicky, eds., *Natural Language Parsing*, 206–250. Cambridge: Cambridge University Press.

Schönfinkel, Moses. 1924. "On the Building Blocks of Mathematical Logic." In Jan van Heijenoort, ed., *From Frege to Gödel*. Harvard University Press, 1967.

Vijay-Shanker, K., and David J. Weir. 1993. "Parsing Some Constrained Grammar Formalisms." *Computational Linguistics*, 19, 591–636.