Optimality is not a Race
Supplementary Handout

Note This handout is pretty much useless without the matching slides, which can be downloaded from the following website: `http://tgraf.bol.ucla.edu/talks.html`

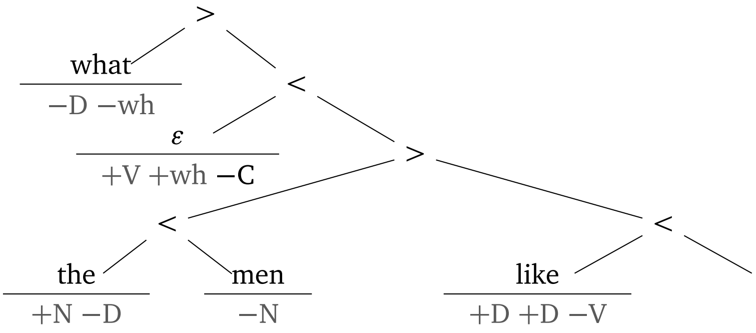
1 Example 1: A Minimalist Grammar

1.1 Lexicon

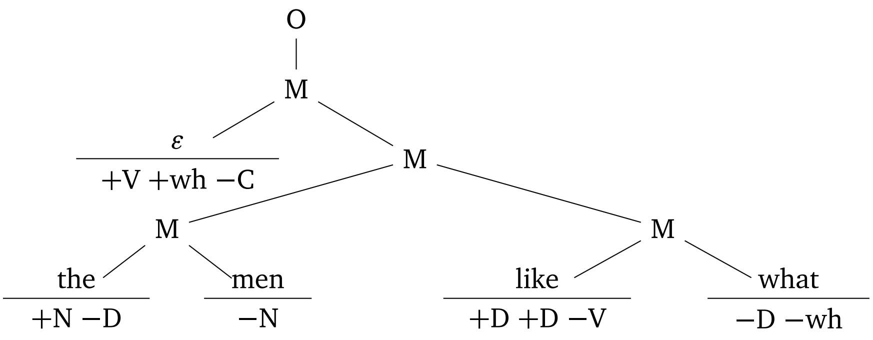
men :: $-$N	like :: $+$D $+$D $-$V
the :: $+$N $-$D	ε :: $+$V ($+$wh) $-$C
what :: $-$D $-$wh	

1.2 Derived Tree

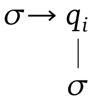
The grayed out parts in the derived tree are only there for the sake of illustration, as checking features is implemented as Erasure in MGs. Consequently, the derived tree contains only the category feature $-$C, all other lexical items are unspecified with respect to their feature-makeup.

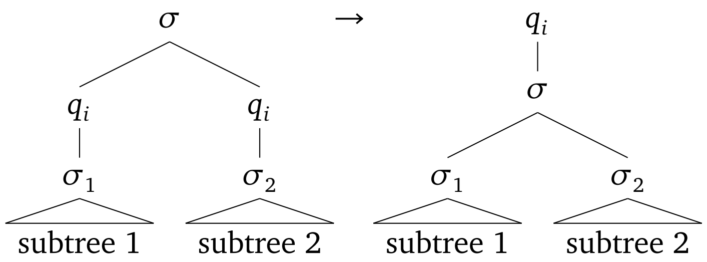


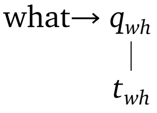
1.3 Derivation Tree

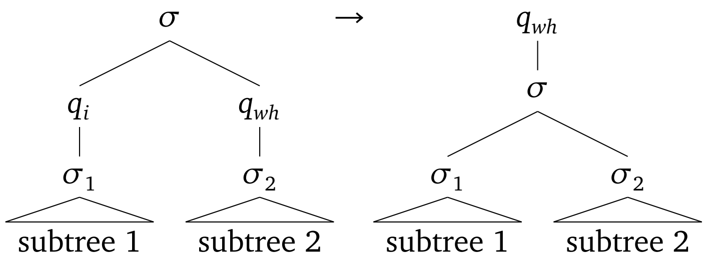


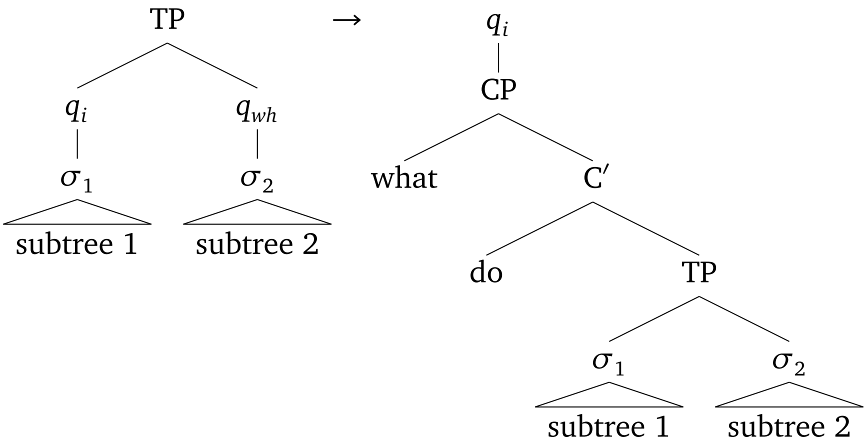
2 Example 2: Transducer for Restricted wh-Movement

(1)


(3)


(2)


(4)


(5)


3 Example 3: Implementing the SDP

3.1 Collins's Empirical Application

(1) a. Who$_i$ did John take [$_{\mathrm{DP}_j}$ a picture of t_i]?
b. * Who$_i$ was [$_{\mathrm{DP}_j}$ a picture of t_i] taken t_j by John?

There are (supposedly) only two derivations for (1b).

- CED violation in (2c) ⇒ ruled out independently

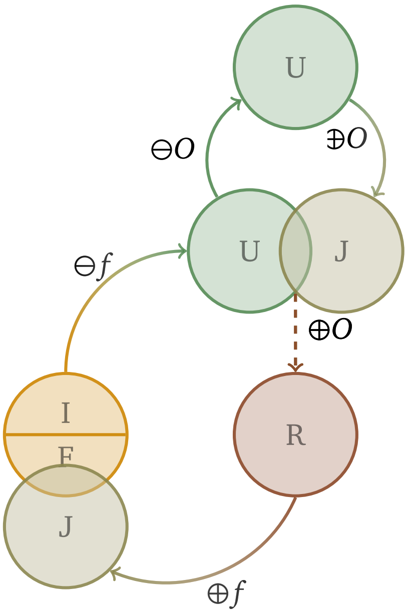
(2) a. [$_{\mathrm{VP}}$ taken [$_{\mathrm{DP}_j}$ a picture of who$_i$] by John]
b. [$_{\mathrm{TP}}$ [$_{\mathrm{DP}_j}$ a picture of who$_i$] T [$_{\mathrm{VP}}$ taken t_j by John]]
c. [$_{\mathrm{CP}}$ who$_i$ was [$_{\mathrm{TP}}$ [$_{\mathrm{DP}_j}$ a picture of t_i] T [$_{\mathrm{VP}}$ taken t_j by John]]]

- longer than (2) ⇒ job for SDP

(3) a. [$_{\mathrm{VP}}$ taken [$_{\mathrm{DP}_j}$ a picture of who$_i$] by John]
b. [$_{\mathrm{VP}}$ who$_i$ taken [$_{\mathrm{DP}_j}$ a picture of t_i] by John]
c. [$_{\mathrm{TP}}$ [$_{\mathrm{DP}_j}$ a picture of t_i] T [$_{\mathrm{VP}}$ who$_i$ taken t_j by John]]
d. [$_{\mathrm{CP}}$ who$_i$ was [$_{\mathrm{TP}}$ [$_{\mathrm{DP}_j}$ a picture of t_i] T [$_{\mathrm{VP}}$ taken t_j by John]]]

3.2 Overview

F	filter
I	input derivations
J	"junk" ≈ overgeneration
R	ranked set
U	underspecified structures
$\ominus f$	remove features
$\oplus f$	add features
$\ominus O$	remove Move nodes
$\oplus O$	add Move nodes



3.3 Step 1: Lexicon Defining Input Language I

John :: −D (−s) (−nom)	by :: +D −P
who :: −N (−s) (−nom) (−wh)	taken :: +D (+P) (+s) −V
a picture of :: +N −D (−s) (−nom)	was :: +V +nom −T
	ε :: +T (+wh) −C

3.4 Step 2: Application of $\ominus f$

We want to preserve the features +/−wh and +/−nom as well as all category features for empirical reasons (we do not want the wh-in-situ construction to compete against the one with wh-movement, and quite generally sentences with different argument structures do not compete) ⇒ only +/−s may be removed
To reduce the complexity of the example, let us assume temporarily that only the following five sentences are deemed well-formed by the grammar (so there have to be extra constraints that rule out all the other possible combinations).

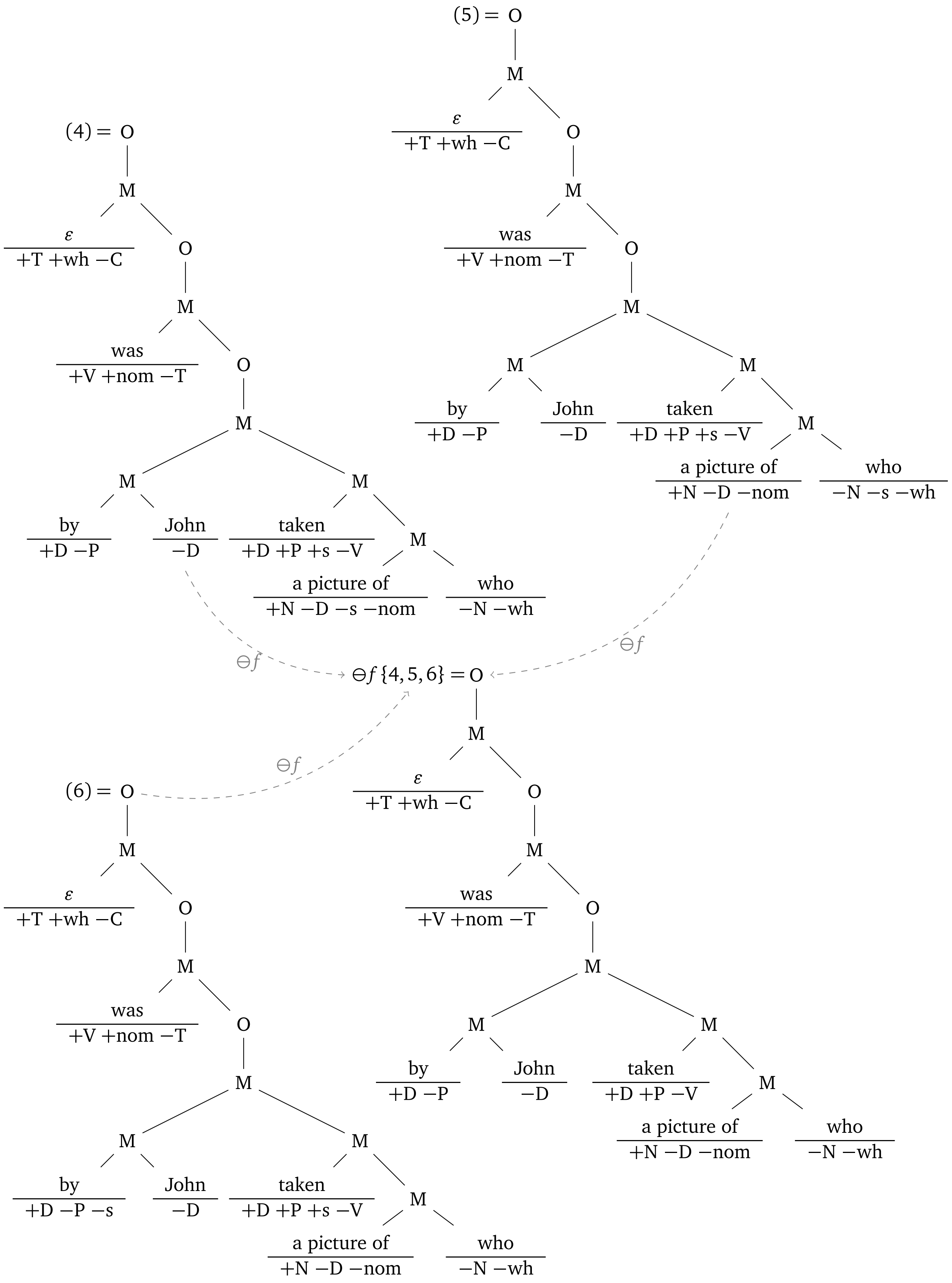
(4) who$_{wh}$ C [$_{\text{DP}}$ a picture of t_{wh}] was [$_{\text{VP}}$ t_{DP} [$_{\text{VP}}$ by John taken t_{DP}]]

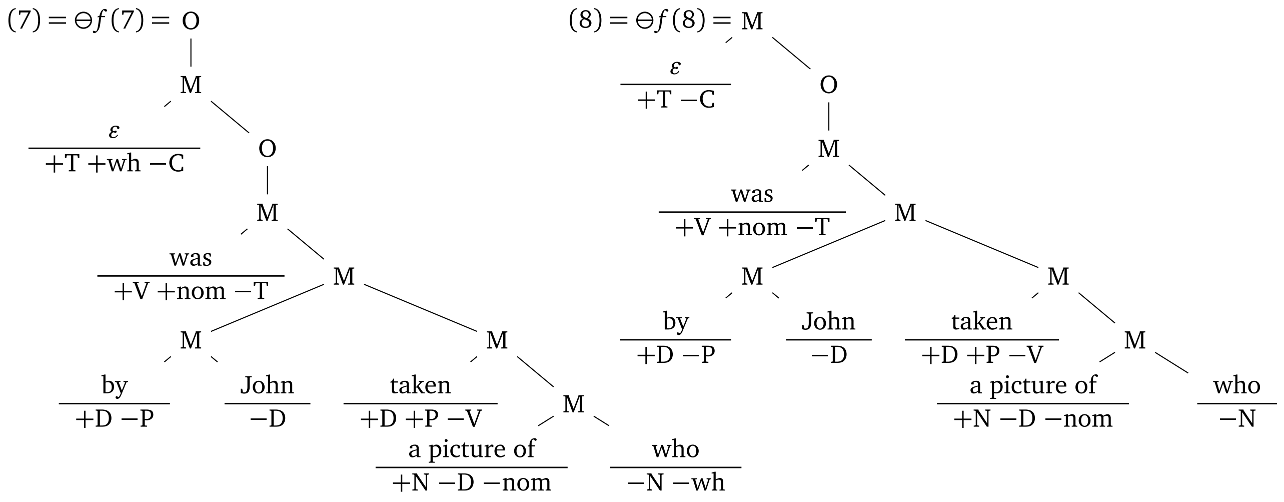
(5) who$_{wh}$ C [$_{\text{DP}}$ a picture of t_{wh}] was [$_{\text{VP}}$ t_{wh} [$_{\text{VP}}$ by John taken t_{DP}]]

(6) who$_{wh}$ C [$_{\text{DP}}$ a picture of t_{wh}] was [$_{\text{VP}}$ [$_{\text{PP}}$ by John] [$_{\text{VP}}$ t_{PP} taken t_{DP}]]

(7) who$_{wh}$ C [$_{\text{DP}}$ a picture of t_{wh}] was [$_{\text{VP}}$ by John taken t_{DP}]

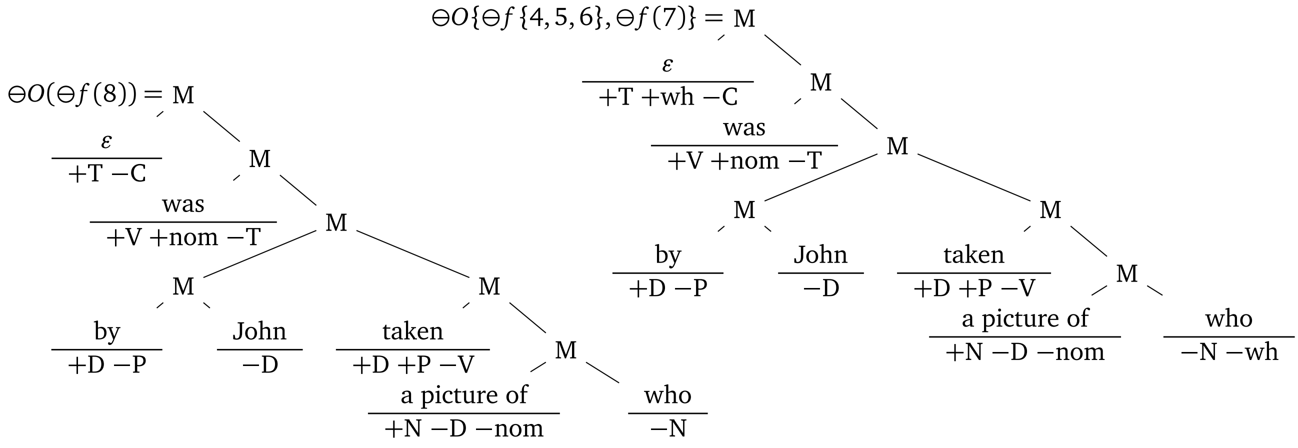
(8) C [$_{\text{DP}}$ a picture of who] was [$_{\text{VP}}$ by John taken t_{DP}]


(5) = O
M
ε
+T +wh −C
O
M
was
+V +nom −T
O
M
M
by
+D −P
John
−D
M
taken
+D +P +s −V
M
a picture of
+N −D −nom
who
−N −s −wh
(4) = O
M
ε
+T +wh −C
O
M
was
+V +nom −T
O
M
M
by
+D −P
John
−D
M
taken
+D +P +s −V
M
a picture of
+N −D −s −nom
who
−N −wh
⊖f
⊖f
⊖f {4, 5, 6} = O
M
ε
+T +wh −C
O
M
was
+V +nom −T
O
M
M
by
+D −P
John
−D
M
taken
+D +P −V
M
a picture of
+N −D −nom
who
−N −wh
⊖f
(6) = O
M
ε
+T +wh −C
O
M
was
+V +nom −T
O
M
M
by
+D −P −s
John
−D
M
taken
+D +P +s −V
M
a picture of
+N −D −nom
who
−N −wh



3.5 Step 3: Application of ⊖O

Applying ⊖O yields two underspecified derivation trees that differ only with respect to the distribution of wh-features.



3.6 Step 4: Application of ⊕O and Filtering

Now we non-deterministically add Movement nodes to each output of ⊖O. This gives us a set of derivation trees, from which we remove any tree that wasn't in the domain of ⊖O; this is simply the mathematical way of ensuring that we do not generate "candidates" that cannot be turned into a derivation licensed by the original grammar. It is easy to see that $\oplus O(\ominus f(8)) = \ominus f(8)$ is disjoint from $\oplus O(\ominus O\{\ominus f\{4,5,6\}, \ominus f(7)\}) = \{\ominus f\{4,5,6\}, \ominus f(7)\}$. So (8) never competes with (4)–(7), as desired. Note that the application of ⊖O followed by ⊕O had no effect on (8), which is still being mapped only to $\ominus f(8)$. Crucially, though, the candidate sets of (4)–(6) on the one hand and (7) on the other are now identical, even though they were previously disjoint.

In summing up: at this point in the transduction, the candidate set of (8) contains only $\ominus f(8)$, whereas (4)–(7) all have the same reference-set, which consists of $\ominus f\{4,5,6\}$ and $\ominus f(7)$.

3.7 Step 5: Ranking via $\oplus O$

We now use $\oplus O$ to define a ranking on the candidates in each reference set. We say that a derivation d is more optimal than d' if d' can be obtained from d by $\oplus O$. $\ominus f(8)$ is trivially optimal, because there is no other tree in the reference set that it could be obtained from via $\oplus O$. For the reference set consisting of $\ominus f\{4,5,6\}$ and $\ominus f(7)$, we see immediately that $\ominus f(7)$ is more optimal, because it differs from the other tree by having one movement node less. Using an idea of Jäger (2002) that exploits some of the properties of regular languages and linear transductions, we transform this ranking into a filter that discards all suboptimal candidate(s).

So the composition of all the mini-transductions encountered so far rewrites (8) as $\ominus f(8)$ and (4)–(7) as $\ominus f(7)$.

3.8 Step 6: Application of $\oplus f$ and Filtering

Remember that we initially stripped away features. To make sure that the end result of the transduction is an actual Minimalist derivation tree licensed by the original grammar, we have to reinstantiate those features. This works like the first time we applied $\oplus O$: Non-deterministically add random material — in this case features — and throw away all undesired outputs. In the case at hand, though, no features have to be reinstantiated since we only stripped away s-features, but no instances of s-movement occur in the trees $\ominus f(7)$ and $\ominus f(8)$ were obtained from.

3.9 Local Correspondent

The local correspondent of the SDP in this case is easy: no lexical item may carry an s-feature. This is true even if we consider all sentences generated by the lexicon above, rather than just (4)–(8). However, the addition of a single lexical item can break this correspondence. Assume, for instance, that the lexicon also contains the lexical item le :: $-\mathrm{D}-\mathrm{s}$, but not le :: $-\mathrm{D}$. Then le always has to move into SpecVP (as indicated by the name, this example is inspired by the behavior of clitics). The local correspondent then is harder to pin down. Intuitively, it would be "don't allow instances of s, unless it cannot be avoided", but this isn't a local constraint. What we need is the following: A verb carries $+\mathrm{s}$ only if it selects le. But this constraint, again, is unstable, since it is too weak for grammars where $+\mathrm{s}$ may occur on other lexical items or le may originate from a lower clause. The SDP, on the other hand, is a principle that applies unaltered in all those cases.

My point is the following: Reference-set constraints can be "precompiled" into local constraints, and thus the two are essentially different ways of stating the same constraint within a grammar, but reference-set constraints afford greater generality by allowing us to formulate constraints that hold across grammars.

References

Jäger, Gerhard. 2002. Gradient constraints in finite state OT: The unidirectional and the bidirectional case. In *More than words. A festschrift for Dieter Wunderlich*, ed. I. Kaufmann and B. Stiebels, 299–325. Berlin: Akademie Verlag.