ACD in a VFS as Binding? W-z isomorphism?

(1) John reads every book that Bill does.

(notational convention: A|B will be used to notate a constituent with a "gap" of cat B, i.e. similarly to A^B)

||does|| = λP[P]	<et,et>	$VP/_{R}VP$
		VP^{VP} (pro-form *does*)
g(||does||) = λRλx[R(x)]	<eet,eet>	$VP^{NP}/_{R}VP^{NP}$
		$(VP^{NP})^{VP\|NP}$ (pro-form)
||Bill$_L$|| = λP[P(b)]	<et,t>	$S/_{R}VP$
g(||Bill$_L$||) = λRλx[R(x)(b)]	<eet,et>	$S^{NP}/_{R}VP^{NP}$
gg(||Bill$_L$||) = λAλRλx[A(R)(x)(b)]	<<eet,eet>,<eet,et>>	$(S^{NP})^{VP\|NP}/_{R}(VP^{NP})^{VP\|NP}$

gg(||Bill$_L$||)(**g**(||does||))

λAλRλx[A(R)(x)(b)](λSλy[S(y)])

λRλx[λSλy[S(y)](R)(x)(b)]

λRλx[(R)(x)(b)] <eet,et> $(S^{NP})^{VP|NP}$

||that|| ignored

||book|| shifts to ||book$_\cap$||

λPλx[book'(x) & P(x)] <et,et> $N/_{R}S^{NP}$

g(||book$_\cap$||) = λAλRλx[book'(x) & A(R)(x)] <<eet,et>,<eet,et>> $N^{VP|NP}/_{R}(S^{NP})^{VP|NP}$

g(||book$_\cap$||)(||Bill does __||)

λAλRλx[book'(x) & A(R)(x)](λSλy[(S)(y)(b)])

λRλx[book'(x) & λSλy[(S)(y)(b)](R)(x)]

λRλx[book'(x) & R(x)(b)] <eet,et> $N^{VP|NP}$

||every$_{pred}$|| = λPλRλx[every'(P)(λy[R(y)(x)])] <et,<eet,et>> $(VP/_{L}(VP/_{R}NP))/_{R}N$

||every$_{pred}$|| ○ ||book that Bill does __||

λPλRλx[every'(P)(λy[R(y)(x)])] ○ λSλz[book'(z) & S(z)(b)]

λSλRλx[every'(λz[book'(z) & S(z)(b)])(λy[R(y)(x)])] $(VP/_{L}(VP/_{R}NP))^{VP|NP}$

Apply **W** to give

λRλx[every'(λz[book'(z) & R(z)(b)])(λy[R(y)(x)])] **(A)**

Apply ||read||

λx[every'(λz[book'(z) & read'(z)(b)])(λy[read'(y)(x)])]

Apply *j*

every'(λz[book'(z) & read'(z)(b)])(λy[read'(y)(j)])

Less formally:

every'(book-that-Bill-reads')(John-reads')

Note that Szabolcsi (1992)'s CCG framework allows us to derive ||every book that Bill does __|| in exactly the same way that Jacobson does, since she has geaching. So, although I've used Jacobson's pro-form notational conventions, this shouldn't be taken as an indication that the derivation is incompatible with Szabolcsi (1992).

It gets more interesting

Instead of **W** we could have done **z**(||every_{pred}||). We would need a generalized **z** rule.

z(λPλRλx[every'(P)(λy[R(y)(x)])])
λfλgλc[f(g(c))(c)](λPλRλx[every'(P)(λy[R(y)(x)])])
λgλc[λPλRλx[every'(P)(λy[R(y)(x)])](g(c))(c)]
λgλcλx[every'(g(c))(λy[c(y)(x)])]

This is of type <<eet,et>,<eet,<et>>>:
λAλRλx[every'(A(R))(λy[R(y)(x)])]

z(||every_{pred}||)(||book that Bill does __||)
λAλRλx[every'(A(R))(λy[R(y)(x)])](λSλz[book'(z) & S(z)(b)])
λRλx[every'(λSλz[book'(z) & S(z)(b)](R))(λy[R(y)(x)])]
λRλx[every'(λz[book'(z) & R(z)(b)])(λy[R(y)(x)])]

Notice the total equivalence of this to **(A)** above:
λRλx[every'(λz[book'(z) & R(z)(b)])(λy[R(y)(x)])] **(A)**

Conclusions (in brief)

ACD is completely compatible with binding analyses. There are reasons we might want this (probably impossible to get any other R-type meaning in the gap) and reasons we might not (the missing meaning can be supplied across a sentence boundary, i.e. *Bagels I like. Donuts I don't.* Of course we could recursively define a category S_{large} composed of individual sentences to get around this... and this might be something we want anyway, given that a series of *n* sentences is well-formed iff each sentence within it is well formed).

This provides independent motivation for a more generally typed **z**.

Composing a function *f* with a constituent *A*|*B* containing a gapped meaning of category *B* and then applying **W** to the result appears to be identical to applying **z** to *f* and then taking *A*|*B* as argument of **z***f*.

References

Jacobson, Pauline. “Direct Compositionality and Variable-Free Semantics: The Case of Antecedent Contained Deletion”, to appear in K. Johnson (ed.), *Topics in Ellipsis*, Oxford University Press.

Szabolcsi, Anna. 1992. Combinatory grammar and projection from the lexicon. In *Lexical matters*, ed. I.A. Sag and A. Szabolcsi, 241-269. Stanford, CSLI Publications.