CONVERGENT GRAMMAR

Carl Pollard
INRIA-Lorraine and Ohio State University

ESSLLI 2008
Hamburg, August 4–8, 2008

The handout for this lecture is available at:

`http://www.ling.ohio-state.edu/~pollard/cvg/day4ho.pdf`

DAY FOUR: INTERROGATIVES

- Interrogative Semantics, Informally
- A Ty2 Review
- Interrogative Semantics, Formally
- Interrogatives in Chinese
- Interrogatives in English

INTERROGATIVE SEMANTICS, INFORMALLY

(1) **Introducing Standard Interrogative Semantics**

- In intensional semantics, the two most influential approaches are due to Karttunen (K, 1997) and to Groenendijk and Stokhof (G, 1984).
- We will review them by looking at how they analyze the **polar interrogative**

  Does Fido bark?/whether Fido barks

  and the **consituent interrogative**

  Who barks?

(2) **K (Karttunen) Questions**

- the reference at w of *whether Fido barks* is the singleton set of propositions whose member is whichever is true at w, that Fido barks, or that he doesn't
- the reference at w of *which dogs bark* is the set of w-true propositions that x barks, where x is a w-dog.
- In both cases, the reference is a **set of propositions**.
- So the questions themselves have the type $\mathrm{s} \rightarrow (\mathrm{s} \rightarrow \mathrm{t}) \rightarrow \mathrm{t}$.

(3) **G (Groenendijk-Stokhof) Questions**

- The reference at w of *whether Fido barks* is the set of worlds each of which agrees with w about which proposition is true there: that Fido barks, or that he doesn't.
- The reference at w of *which dogs bark* is the set of worlds which agree, for each individual x, about which proposition is true there: that x is a dog that barks, or that it isn't.
- In both cases, the reference is a **set of worlds**.
- So the questions themselves have the type

  $\mathrm{s} \rightarrow \mathrm{s} \rightarrow \mathrm{t}$

  (the type of binary relations on worlds—in fact they are equivalence relations).

(4) **$K_{\pm}$ (Plus-or-Minus Karttunen) Questions**

- G-questions have one advantage over K-questions: they take **both positive and negative** 'true atomic answers' into account, **for both polar and constituent questions**, while K-questions **throw out the negative answers** in the case of constituent questions.
- Pollard 2008b (following a suggestion by Ken Shan) proposes a modification of K-questions that incorporate this feature.
- Polar questions are handled the same as in K.
- Constituent questions are handled as in K, except **both positive and negative true atomic answers** are kept.
- So the reference of *which dog barked* is the set of all w-true propositions that x barks **or** that x **doesn't** bark, for x a w-dog.

(5) **More on Plus-or-Minus Karttunen Questions**

- Any function induces an equivalance relation on its domain, the relation of being mapped to the same value
- Since $K_{\pm}$-question are of type s → (s → t) → t, each one induces an equivalence relation on the set of worlds.
- In fact, it is just the corresponding G-question.
- In general, going from a function to the induced equivalence relation losed information: the mapping is not injective.
- For example:

  Which students are vegetarians?

  Which vegetarians are students?

  yield distinct $K_{\pm}$-questions, but the same G-question.
- The interrogative semantics we will actually use is a slight elaboration of the $K_{\pm}$-semantics.

A TY2 REVIEW

(6) **Ty2**

- Ty2 is a classical higher-order logic built on top of TLC.
- More specifically, it is Henkin's (1950) axiomatization of Church's (1940) simple theory of types, with one additional basic type.
- In standard linguistic notation, the basic types are e (entities), t (truth values), and s (worlds).

(7) **Ty2 Meaning Types**

These are defined as follows:

a. s $\rightarrow$ e (individual concepts) is a Ty2 meaning type.

b. s $\rightarrow$ t (propositions) is a Ty2 meaning type.

c. If A and B are Ty2 meaning types, then so is $A \rightarrow B$.

(8) **Basic Types of CVG Semantic Logic**

- For present purposes, we use these three:

  ι (individual concepts)

  π (propositions)

  and κ (polar questions).

- We don't presuppose a specific semantic theory, but as a practical matter, in this course we'll interpret CVG semantic terms by translating them into Ty2 (whose semantics is well known).
- We'll elaborate our τ-transform into TLC to map our basic semantic types to basic meaning types of Ty2.
- We could just as well have usqed **hyperintensional** semantics for this purpose.

(9) **Notation for Functional Types (Review)**

Where σ ranges over strings of types and ϵ is the null string:

i. $A_\epsilon =_{\text{def}} A$

ii. $A_{B\sigma} =_{\text{def}} B \to A_\sigma$ (e.g. $\pi_{\iota\iota} = \iota \to \iota \to \pi$)

iii. For $n \in \omega$, $A_n =_{\text{def}} A_\sigma$ where σ is the string of ι's of length n

Example: $\pi_2 =_{\text{def}} \pi_{\iota\iota} =_{\text{def}} \iota \to \iota \to \pi$.
Note: This clunky notation is the price we pay for not having conjunction in the type logic.

(10) **The Transform τ from CVG Types to Ty2 Meaning Types**

a. $\tau(\iota) = \mathrm{s} \to \mathrm{e}$

b. $\tau(\pi) = \mathrm{s} \to \mathrm{t}$

c. $\tau(\kappa) = \tau(\pi) \to \tau(\pi)$

d. $\tau(A \to B) = \tau(A) \to \tau(B)$

e. $\tau(A_B^C) = (\tau(A) \to \tau(B)) \to \tau(C)$

(11) **The Transform τ on Terms (Review)**

a. Variables and basic constants are unchanged except for their types.

b. $\tau((f\ a)) = \tau(f)(\tau(a))$

The change in the parenthesization has no theoretical significance. It just enables one to tell at a glance whether the term belongs to the CVG semantic logic or to Ty2, e.g. (walk' Kim') vs. walk'(Kim').

c. $\tau((a_x b)) = \tau(a)(\lambda_x \tau(b))$

Note: This is the important clause. It says that operator binding consists of abstraction immediately followed by application.

(12) **Ty2 Meaning Postulates for Generalized Quantifiers**

$\vdash \textsf{every'} = \lambda_Q \lambda_P \lambda_w \forall_x (Q(x)(w) \rightarrow P(x)(w))$

$\vdash \textsf{some'} = \lambda_Q \lambda_P \lambda_w \exists_x (Q(x)(w) \wedge P(x)(w))$

$\vdash \textsf{everyone'} = \textsf{every'}(\textsf{person'})$

$\vdash \textsf{someone'} = \textsf{some'}(\textsf{person'})$

Types for Ty2 variables are as follows:

$x, y, z : \mathrm{s} \rightarrow \mathrm{e}$ (individual concepts)

$p, q : \mathrm{s} \rightarrow \mathrm{t}$ (propositions); $w : \mathrm{s}$ (worlds)

$P, Q : ((\mathrm{s} \rightarrow \mathrm{e}) \rightarrow (\mathrm{s} \rightarrow \mathrm{t}))$ (properties of individual concepts).

(13) **Extensional Types Corresponding to Ty2 Meaning Types**

These are defined recursively as follows:

a. $\mathsf{E}(\mathrm{s} \rightarrow \mathrm{e}) = \mathrm{e}$

b. $\mathsf{E}(\mathrm{s} \rightarrow \mathrm{t}) = \mathrm{t}$

c. $\mathsf{E}(A \rightarrow B) = A \rightarrow \mathsf{E}(B)$, where A and B are Ty2 meaning types.

(14) **Extensions of Ty2 Meanings**

The relationship between Ty2 meanings and their extensions is axiomatized as follows, where the family of constants $\mathsf{ext}_A : \mathrm{s} \to A \to \mathsf{E}(A)$ is parametrized by the Ty2 meaning types:

a. $\vdash \forall_x \forall_w (\mathsf{ext}_w(x) = x(w)$ (for $x : \mathrm{s} \to \mathrm{e}$)

b. $\vdash \forall_p \forall_w (\mathsf{ext}_w(p) = p(w)$ (for $p : \mathrm{s} \to \mathrm{t}$)

c. $\vdash \forall_f \forall_w (\mathsf{ext}_w(f) = \lambda_x \mathsf{ext}_w(f(x))$ (for $f : A \to B$, A and B Ty2 meaning types.

Note: we suppress the type parameter, and write ext_w for $\mathsf{ext}(w)$.

INTERROGATIVE SEMANTICS, FORMALLY

(15) **Overall Approach to Interrogative Semantics**

The approach is described in detail in Pollard 2008. Key ideas:

- The analysis of polar questions (after transformation into Ty2) is that of Karttunen 1977: at each world w, an interrogative sentence denotes a set of w-facts (in this case, a singleton).
- For n-ary constituent interrogatives, the denotation at w is a (curried) n-ary function to w-facts. The range of that function is similar to the Karttunen semantics, except that it contains both positive and negative 'true atomic answers.'
- An interrogative meaning of this kind induces an equivalence relation on worlds which is a **refinement** of the Groenendijk-Stokhof (1984) partition semantics.

(16) **Types for Polar Questions**

a. CVG meaning type: κ

b. Meaning type of Ty2 transform: $(s \to t) \to (s \to t)$ (property of propositions)

c. Type of Ty2 denotation: $(s \to t) \to t$ (set of propositions)

d. Example: at w, *Does Chris walk* (or *whether Chris walks*) denotes the singleton set whose member is whichever is true at w, the proposition that Chris walks or the proposition that s/he doesn't.

(17) **Types for Unary Constituent Questions**

a. CVG meaning type: κ_1

b. Meaning type of Ty2 transform: (s $\rightarrow$ e) $\rightarrow$ (s $\rightarrow$ t) $\rightarrow$ (s $\rightarrow$ t) (function from individual concepts to properties of propositions).

c. Type of Ty2 denotation: (s $\rightarrow$ e) $\rightarrow$ (s $\rightarrow$ t) $\rightarrow$ t (function from individual concepts to sets of propositions).

d. Example: at w, *who walks* denotes the (functional) binary relation between individual concepts x and propositions p that obtains just in case x is a w-person and and p is whichever proposition is a w-fact, that x walks or that x does not walk.

(18) **Types for Binary Constituent Questions**

a. CVG meaning type: κ_2

b. Meaning type of Ty2 transform: $(s \rightarrow e) \rightarrow (s \rightarrow e) \rightarrow (s \rightarrow t) \rightarrow (s \rightarrow t)$ (curried function from pairs of individual concepts to properties of propositions).

c. Type of Ty2 denotation: $(s \rightarrow e) \rightarrow (s \rightarrow e) \rightarrow (s \rightarrow t) \rightarrow t$ (curried function from pairs of individual concepts to sets of propositions).

d. Example: at w, *who likes what* denotes the (functional) ternary relation between individual concepts x and y and propositions p that obtains just in case x is a w-person, y is a w-thing, and p is whichever proposition is a w-fact, that x likes y or that x does not like y.

(19) **Types for Interrogatives (Summary)**

a. The CVG type for polar interrogatives (*whether Fido barked*) is $\kappa_0 = \kappa$, whose Ty2 transform is $(\mathrm{s} \rightarrow \mathrm{t}) \rightarrow \mathrm{s} \rightarrow \mathrm{t}$ (property of propositions).

b. The CVG type for unary constituent interrogatives (*who barked*) is $\kappa_1 = \iota \rightarrow \kappa$, whose Ty2 transform is $\iota \rightarrow (\mathrm{s} \rightarrow \mathrm{t}) \rightarrow \mathrm{s} \rightarrow \mathrm{t}$ (function from individuals to properties of propositions).

c. The CVG type for binary consituent interrogatives (*who bit who*) is $\kappa_2 = \iota \rightarrow \iota \rightarrow \kappa$, whose Ty2 transform is $\iota \rightarrow \iota \rightarrow (\mathrm{s} \rightarrow \mathrm{t}) \rightarrow \mathrm{s} \rightarrow \mathrm{t}$ (curried function from pairs of individuals to properties of propositions), etc.

(20) **Multiple *Wh*-In Situ vs. Multiple Quantifier Raising**

a. The fact that not all questions have the same type introduces a complexity that does not arise with quantifier scope.

b. But as we'll see, it also explains s lot.

b. Since the result type of a quantifier is the same as its scope type, we can scope multiple quantifiers one after the other.

c. But (for example,) scoping one (overtly moved) *wh*-operator at a proposition produces a unary consituent question, so its type must be $\iota_{\pi}^{\kappa_1}$.

d. So if we want to scope a second (in-situ) *wh*-operator over that unary constituent question to form a binary consituent question, then *its* type must be $\iota_{\kappa_1}^{\kappa_2}$, etc.

d. We will return to this point presently.

(21) **Ty2 Meaning Postulates for Standard Logical Constants**

a. $\vdash \mathsf{id}_A$

b. $\vdash \mathsf{and'} = \lambda_p \lambda_q \lambda_w (p(w) \wedge q(w))$

c. $\vdash \mathsf{or'} = \lambda_p \lambda_q \lambda_w (p(w) \vee q(w))$

d. $\vdash \mathsf{not'} = \lambda_p \lambda_w \neg p(w)$

e. $\vdash \mathsf{equals'}_A = \lambda_x \lambda_y \lambda_w (x = y)$

(22) **Ty2 MPs for Some Less Standard Logical Constants**

a. $\vdash \mathsf{whether'} = \lambda_q \lambda_p (p \text{ } \mathsf{and'} \text{ } ((p \text{ } \mathsf{equals'} \text{ } q) \text{ } \mathsf{or'} \text{ } (p \text{ } \mathsf{equals'} \text{ } \mathsf{not'}(q))))$

b. $\vdash \mathsf{which}^0 = \lambda_Q \lambda_P \lambda_x \lambda_p (Q(x) \text{ } \mathsf{and'} \text{ } \mathsf{whether'}(P(x))(p))$

c. $\vdash \mathsf{which}^n = \lambda_Q \lambda_Z \lambda_{x_0} \ldots \lambda_{x_n} \lambda_p (Q(x) \text{ } \mathsf{and'} \text{ } Z(x_0) \ldots (x_n)(p)) \; (n > 0)$

d. $\vdash \mathsf{who}^n = \mathsf{which}^n(\mathsf{person'})$

e. $\vdash \mathsf{what}^n = \mathsf{which}^n(\mathsf{thing'})$

INTERROGATIVES IN CHINESE

(23) **Types, Categories, and Schemata for Chinese Fragment**

The same as for English (for this fragment anyway).

This improves on the analysis of Pollard 2007a,b which required construction-specific rules for different kinds of in-situ operators.

(24) **Lexicon for Chinese Fragment**

$\vdash$ Zhangsan, Zhangsan' : $\mathrm{NP}, \iota \dashv$

$\vdash$ xihuan, like' : $(\mathrm{NP} \multimap_{\mathrm{C}} (\mathrm{NP} \multimap_{\mathrm{S}} \mathrm{S}), \iota \rightarrow (\iota \rightarrow \pi) \dashv$

$\vdash$ xi-bu-xihuan, like?' : $(\mathrm{NP} \multimap_{\mathrm{C}} (\mathrm{NP} \multimap_{\mathrm{S}} \mathrm{S}), \iota \rightarrow (\iota \rightarrow \kappa) \dashv$

$\vdash$ xiang-zhidao, wonder'$_n$: $\mathrm{S} \multimap_{\mathrm{C}} (\mathrm{NP} \multimap_{\mathrm{S}} \mathrm{S}), \kappa_n \rightarrow (\iota \rightarrow \pi) \dashv$

$\vdash$ shei, who^0 : $\mathrm{NP}, \iota_{\pi}^{\kappa_1} \dashv$

$\vdash$ shei, who^n : $\mathrm{NP}, \iota_{\kappa_n}^{\kappa_{n+1}} \dashv$ (for $n > 0$)

$\vdash$ shenme, what^0 : $\mathrm{NP}, \iota_{\pi}^{\kappa_1} \dashv$

$\vdash$ shenme, what^n : $\mathrm{NP}, \iota_{\kappa_n}^{\kappa_{n+1}} \dashv$ (for $n > 0$)

(25) **Meaning Postulate for an Interrogative Verb Meaning**

$\vdash$ like?' $= \lambda_y \lambda_x$whether'(like'$(y)(x)$)

(26) **Comments on the Chinese Lexicon**

- *xibuxihuan* 'like?' is a partial-reduplicative interrogative verb form, used for forming (both root and embedded) polar questions.
- The meaning of *xiang-zhidao* 'wonder' has to be type-schematized according to the type of question expressed by the sentential complement.
- The meanings of the *sh*-interrogative words have to be type-schematized according to their scope type (and corresponding result type).

(27) **A Simple Chinese Sentence**

a. Zhangsan xihuan Lisi.

b. Zhangsan like Lisi

c. 'Zhangsan likes Lisi.'

d. $\vdash$ ($^{\mathrm{S}}$ Zhangsan (xihuan Lisi $^{\mathrm{C}}$)) : S

e. Ty2: $\vdash$ like'(Lisi')(Zhangsan') : $\tau(\pi)$

(28) **A Chinese Polar Question**

a. Zhangsan xi-bu-xihuan Lisi?

b. Zhangsan like? Lisi

c. 'Does Zhangsan like Lisi?'

d. $\vdash$ ($^{\mathrm{S}}$ Zhangsan (xi-bu-xihuan Lisi $^{\mathrm{C}}$)) : S

e. Ty2: $\vdash$ whether'(like'(Lisi')(Zhangsan')) : $\tau(\kappa_0)$

(29) **A Chinese Unary Constituent Question**

a. Zhangsan xihuan shenme?

b. Zhangsan like who

c. 'What does Zhangsan like?'

d. $\vdash$ ($^{\mathrm{s}}$ Zhangsan (xihuan shenme $^{\mathrm{c}}$)) : S

e. Sem: $\vdash$ (what^0_y(($\mathsf{like'}$ y) ($\mathsf{Zhangsan'}$)) : κ_1 $\dashv$

(30) **A Chinese Binary Constituent Question**

a. Shei xihuan shenme?

b. who like what

c. ‘Who likes what?’

d. $\vdash$ ($^{\mathrm{s}}$ Shei (xihuan shenme $^{\mathrm{c}}$)) : S

e. Sem: $\vdash (\mathsf{who}^1_x(\mathsf{what}^0_y((\mathsf{like'}\ y)\ (x))) : \kappa_2 \dashv$

f. Sem: $\vdash (\mathsf{what}^1_y(\mathsf{who}^0_x((\mathsf{like'}\ y)\ (x))) : \kappa_2 \dashv$

The ambiguity is inessential: the two functions are the same modulo permutation of their arguments.

(31) **Baker-Type Ambiguity in English**

a. A: Who knows where we bought what?

b. B: Chris does. (*what* scopes to embeded question.)

c. B: Chris knows where we bought the books, and Kim knows where we bought the records. (*what* scopes to root question.)

d. The syntactic operators *who* and *where* must scope at their 'surface' positions.

e. But the in-situ *what* can scope high or low.

(32) **A Chinese Baker Ambiguity**

a. Zhangsan xiang-zhidao shei xihuan shenme./?

b. Zhangsan wonder who like what

c. $\vdash$ ($^{\mathrm{S}}$ Zhangsan (xiang-zhidao ($^{\mathrm{S}}$ shei (xihuan shenme $^{\mathrm{C}}$) $^{\mathrm{C}}$))) : S

d. $\vdash$ ((wonder'$_2$ (who^1_x(what^0_y((like' y) x)))) Zhangsan') : $\pi \dashv$
'Zhangsan wonders who likes what.'

e. $\vdash$ (who^0_x((wonder'$_1$ (what^0_y((like' y) x))) Zhangsan') : $\kappa_1 \dashv$
'Who does Zhangsan wonder what (that person) likes?'

f. $\vdash$ (what^0_y((wonder'$_1$ (who^0_x((like' y) x))) Zhangsan') : $\kappa_1 \dashv$
'What does Zhangsan wonder who likes?'

(33) **The Gist of the Preceding**

a. Both *sh*-expressions are in situ, so they can each scope high or low.

b. If both scope low (32), then the root sentence expresses a proposition and the embedded sentence expresses a binary question.

c. If one scopes high and the other low (32,32), then the root sentence and the embedded sentence both express unary questions.

d. But they cannot *both* scope high, since then the complement sentence would express a proposition, while the first argument of wonder′ must be a question.

(34) **Lexicon for Interrogative Fragment**

$\vdash$ Kim, Kim' : $\mathrm{NP}, \iota \dashv$

$\vdash$ liked, like' : $\mathrm{NP} \multimap_{\mathrm{C}} \mathrm{NP} \multimap_{\mathrm{S}} \mathrm{S}, \iota \rightarrow \iota \rightarrow \pi \dashv$

$\vdash$ whether, whether' : $\mathrm{S} \multimap_{\mathrm{M}} \mathrm{S}, \pi \rightarrow \kappa \dashv$

$\vdash$ wondered, wonder'$_n$: $\mathrm{S} \multimap_{\mathrm{C}} \mathrm{NP} \multimap_{\mathrm{S}} \mathrm{S}, \kappa_n \rightarrow \iota \rightarrow \pi \dashv$

$\vdash$ who$_{\text{filler}}$, who^0 : $\mathrm{NP}^{\mathrm{Q}}_{\mathrm{S}}, \iota^{\kappa_1}_{\pi} \dashv$

$\vdash$ who$_{\text{in-situ}}$, who^n : $\mathrm{NP}, \iota^{\kappa_{n+1}}_{\kappa_n} \dashv$ (for $n > 0$)

$\vdash$ what$_{\text{filler}}$, what^0 : $\mathrm{NP}^{\mathrm{Q}}_{\mathrm{S}}, \iota^{\kappa_1}_{\pi} \dashv$

$\vdash$ what$_{\text{in-situ}}$, what^n : $\mathrm{NP}, \iota^{\kappa_{n+1}}_{\kappa_n} \dashv$ (for $n > 0$)

(35) **Comments on the English Lexicon**

- Unlike Chinese, (embedded) polar interrogatives are formed by 'complementizing' declarative sentences with *whether*, which has the same meaning as the Chinese interrogative reduplicative verbal affix.
- The verb *wonder* is type-schematized according to the type of question expressed by the sentential complement.
- The meanings of the *wh*-interrogative words are type-schematized as in Chinese, with one crucial difference.

(36) **Observations about Interrogative *who***

- The interrogative 'pronoun' *who* is **syntactically** ambiguous between a syntactic operator $\mathsf{who}_{\mathsf{filler}}$ and an NP, $\mathsf{who}_{\mathsf{in\text{-}situ}}$.
- $\mathsf{who}_{\mathsf{filler}}$ can only form an interrogative (Q) by scoping syntactically over a 'declarative' (i.e. semantically propositional) S containing at least one unbound NP trace, and the semantic result (formed by scoping who^0 over an open proposition), is a unary constituent question (type κ_1).
- $\mathsf{who}_{\mathsf{in\text{-}situ}}$ cannot scope syntactically, but its stored meaning (any of who^n, $n > 0$) can be retrieved at a constituent question (type κ_n, $n > 0$) to form a 'higher' consituent question (type κ_{n+1}).

(37) **Consequences**

- There can be no purely in-situ interrogatives (leaving aside pragmatically restricted, intonationally marked ones which we cannot go into here):

  **I wonder Fido bit who?*

- A *wh*-expression cannot scope, either overtly or covertly, over a polar interrogative:

  **I wonder whether Fido bit who?*

  **I wonder who whether Fido bit?*

- In each constituent interrogative, only one 'overtly moved' *wh*-expression can take scope there:

  **I wonder who who(m) bit?*

(38) **More Consequences**

- Arbitrarily many in-situ *wh*-expressions can take their semantic scope at a given consituent interrogative:

  Who gave what to who when?

- There are (Baker) ambiguities that hinge on how high an in-situ *wh*-expression scopes:

  Who wondered who bit who?

- Even though subject *wh*-expressions might *look* in situ:

  Who barked?

  they aren't really; if they were, they could also scope higher to form imposssible embedded questions as in:

  **Kim wondered Chris thought who barked*?

  (Intended meaning: Kim wondered who Chris thought barked.)

(39) **Wh-In Situ Languages**

In languages without overt *wh*-movement, the counterpart of *who* is just an NP with **all** the meanings who^n $(n \geq 0)$, **including** who^0.

That is: the difference between overt and covert *wh*-movement languages is in the lexicon.

(40) **An English Embedded Polar Question**

a. Syntax: $\vdash$ (whether ($^{\mathrm{S}}$ Kim (likes Sandy $^{\mathrm{C}}$)) $^{\mathrm{C}}$) : S

b. Semantics: $\vdash$ (whether' (like' Sandy' Kim')) : κ_0

(41) **An English Embedded Constituent Question**

a. Syntax: $\vdash$ $[\mathsf{what}_{\mathsf{filler}}\ {}_{t}({}^{\mathrm{S}}\ \mathsf{Kim}\ (\mathsf{likes}\ t\ {}^{\mathrm{C}}))]$: S

b. Semantics: $\vdash$ $(\mathsf{what}^{0}_{y}((\mathsf{like'}\ y)\ (\mathsf{Kim'}))$: κ_1

(42) **A English Binary Constituent Question**

a. Syntax: $\vdash$ $[\mathsf{who}_{\mathsf{filler}}\ {}_{t}({}^{\mathrm{S}}\ t\ (\mathsf{likes}\ \mathsf{what}_{\mathsf{in\text{-}situ}}\ {}^{\mathrm{C}}))]$: S

b. Semantics: $\vdash$ $(\mathsf{what}^{1}_{\underline{y}}(\mathsf{who}^{0}_{x}((\mathsf{like'}\ \underline{y})\ (x)))$: κ_2

(43) **Baker-Type Ambiguity in English (Review)**

a. A: Who wonders who likes what?

b. B: Chris does. (Appropriate when *what* scopes to the embeded question.)

c. B: Chris wonders who likes the books, and Kim wonders who likes the records. (Appropriate when *what* scopes to the root question.)

d. The 'overtly moved' *wh*-expressions must scope at their 'surface' positions: *who* can only scope to the root question, and *where* can only scope to the embedded question.

   Now we know why: because syntactic operators (fillers in Schema G) are interpreted 'in place'.

e. But the in-situ *wh*-expression *what* can scope high or low.

   Now we know why: in-situ operators are stored, and have multiple options as to where they are retrieved.

(44) **Baker Ambiguity in English**

a. $\vdash$ [$\textsf{who}_{\textsf{filler}}\ {}_t$($^{\text{S}}$ t ($\textsf{wonders}$ [$\textsf{who}_{\textsf{filler}}\ {}_{t'}$($^{\text{S}}$ t' ($\textsf{likes}$ $\textsf{what}_{\textsf{in-situ}}$ $^{\text{C}}$))] $^{\text{C}}$))] : S

b. $\vdash (\textsf{who}^0_x((\textsf{wonder'}_2\ (\textsf{what}^1_{\underline{y}}(\textsf{who}^0_z((\textsf{like'}\ \underline{y})\ z))))\ x)) : \pi$

(E.g. Chris wonders who likes what.)

c. $\vdash (\textsf{what}^1_{\underline{y}}(\textsf{who}^0_x((\textsf{wonder'}_1\ (\textsf{who}^0_z((\textsf{like'}\ \underline{y})\ z)))\ x)))) : \pi$

(E.g. Chris wonders who likes the books, and Kim wonders who likes the records.)

(45) **The Gist of the Preceding**

a. Both instances of *who* are fillers, so they must scope ‘in place’.

b. But *what* is in situ, so it can scope high or low.

(46) **The Difference between Chinese and English**

- The main difference between English and Chinese, at least as far as interrogatives are concerned, is a lexical one: in English, but not in Chinese, the interrogative pronouns are syntactically ambiguous.
- For example, there is only one Chinese syntactic word *shei* 'who', namely shei : NP.
- But in English, the *who* with meaning $\mathsf{who}^0 : \iota^{\kappa}_{\pi}$, the one that 'gets the ball rolling', is **not** an NP but rather the syntactic operator $\mathsf{who}_{\mathsf{filler}} :^{\mathrm{S}}_{\mathrm{S}} \mathrm{NP})$.
- This is the wrong category to be a subject or a complement, but the right category to be a filler in Schema G.
- On the other hand, the English *who* that expresses all the all the other meanings ($\mathsf{who}_n : \iota^{\kappa_{n+1}}_{\kappa_n}$, for $n > 0$), namely $\mathsf{who}_{\mathsf{in\text{-}situ}}$, is an NP just like Chinese shei, so it is always in situ.