Lecture 3
SEMANTIC COMPOSITION

1. Combinatory Categorial Grammar (CCG)

BASIC IDEA (Steedman 2000):
The syntax and semantics of nat. lgs. can be specified as for TL, i.e. a CCG grammar consists of:
i. *lg-specific* **lexicon** that for each item specifies a category & basic meaning (e.g. $\text{he}_n \mid\!- \text{np}_\text{N}: z_n$)
CCG categories are either basic (e.g. s, np) or complex (like TL types). The latter are functor categories (e.g. $(\text{s}\backslash\text{np}_\text{N})/\text{np}_\text{A}$ for English transitive verbs) that specify:
- categories of the arguments (accusative noun phrase, np_A, and nominative, np_N)
- order of combination ($(\text{s}\backslash\text{np}_\text{N})/\text{np}_\text{A}$ combines first with np_A, then with np_N)
- linear order if grammaticalized ($(\text{s}\backslash\text{np}_\text{N})/\text{np}_\text{A}$ looks for np_A to the right, np_N to the left)
- category of the result (sentence, s)[1]

CCG categories are language-specific, except for the universal category s (sentence).
ii. *universal* **combinatory rules** combine lexical items, based on their categories and meanings, and determine the category as well as the meaning of the result (like TL rules D1.3 and D2.3)
CCG, like other categorial grammars, builds on the so-called *AB calculus* (see below), a core system of categories and rules developed by Ajdukiewicz (1935) and Bar-Hillel (1953).

AB CATEGORIES
Given $\mathcal{A}$, a finite set of atomic categories, the set $\mathcal{C}$ of categories is the smallest set such that:
i. $\mathcal{A} \subseteq \mathcal{C}$
ii. $(X\backslash Y), (X/Y) \in \mathcal{C}$ if $X, Y \in \mathcal{C}$

AB RULES
(>) *Forward application*
$(X/Y) : f \quad Y : a \quad \Rightarrow \quad X : f(a)$
[A constituent of category (X/Y), denoting a function f, combines with a directly following constituent of category Y, denoting an argument $a \in \text{Dom } f$, into a constituent of category X, denoting $f(a)$.]

(<) *Backward application*
$Y : a \quad (X\backslash Y) : f \quad \Rightarrow \quad X: f(a)$
[A constituent of category (X\Y), denoting a function f, combines with a directly preceding constituent of category Y, denoting an argument $a \in \text{Dom } f$, into a constituent of category X, denoting $f(a)$.]

ABBREVIATIONS FOR TL-TERMS (used to represent meanings)
- Parentheses and brackets can be omitted if they can be unambiguously restored by D1.3, e.g.
$x = z_2 \wedge Rz_2z_3 \quad := \quad ((x = z_2) \wedge Rz_2z_3)$
- λu with missing brackets takes widest possible scope, e.g.
$\lambda x\, x \quad := \quad \lambda x[x]$
$\lambda y[\lambda x\, see_{e(et)}\, yx]z \quad := \quad \lambda y[\lambda x[see_{e(et)}\, yx]]z$
$\lambda y\lambda x[see_{e(et)}\, yx]z \quad := \quad \lambda y[\lambda x[see_{e(et)}\, yx]z]$

[1] We follow the *result leftmost* convention (of Steedman 2000), rather than the *result topmost* convention (of Lambek calculus), where the corresponding category would be $(\text{np}_\text{N}\backslash\text{s})/\text{np}_\text{A}$.

2. English fragment 1: CCG + TL

In the following fragment of English we use TL-terms to name the semantic objects (i.e., entities, truth values, or functions) that English items denote and the AB rules combine.

NOTATION: X ranges over *cases*: A = accusative (object case), N = nominative (subject case)

Eng. item	|–	AB category: TL-translation
he_n	|–	np_{N}: z_n
him_n	|–	np_{A}: z_n
Jim_n	|–	np_{X}: $\iota x(x = z_n \wedge z_n = jim)$
$\text{his}_{m,n}$	|–	$\text{np}_{\text{X}}/\text{n}$: $\lambda R[\iota x(x = z_n \wedge Rz_m z_n)]$
$\text{Jim}_m\text{'s}_n$	|–	$\text{np}_{\text{X}}/\text{n}$: $\lambda R[\iota x(x = z_n \wedge (Rz_m z_n \wedge z_m = jim))]$
husband	|–	n: $hsb_{e(et)}$
to	|–	$\text{np}_{to}/\text{np}_{\text{A}}$: $\lambda x[x]$
came.in	|–	$\text{s}\backslash\text{np}_{\text{N}}$: cm_{et}
come.in	|–	$\text{s}_{inf}\backslash\text{np}_{\text{N}}$: cm_{et}
saw	|–	$(\text{s}\backslash\text{np}_{\text{N}})/\text{np}_{\text{A}}$: $see_{e(et)}$
see	|–	$(\text{s}_{inf}\backslash\text{np}_{\text{N}})/\text{np}_{\text{A}}$: $see_{e(et)}$
spoke	|–	$(\text{s}\backslash\text{np}_{\text{N}})/\text{np}_{to}$: $spk_{e(et)}$
speak	|–	$(\text{s}_{inf}\backslash\text{np}_{\text{N}})/\text{np}_{to}$: $spk_{e(et)}$
didn't	|–	$(\text{s}\backslash\text{np}_{\text{N}})/(\text{s}_{inf}\backslash\text{np}_{\text{N}})$: $\lambda P[\lambda x[\neg Px]]$
⋮		

(the ellipsis stands for other English pronouns, names, relational nouns, case marking prepositions, intransitive verbs, and transitive verbs, with analogous lexical entries.)

3. Sample compositional analyses

(1) Jim_1 came.in

Jim_1: np_{N}: $\iota x(x = z_1 \wedge z_1 = jim)$

came.in: $\text{s}\backslash\text{np}_{\text{N}}$: cm_{et}

$<$ s: $cm_{et}\ \iota x(x = z_1 \wedge z_1 = jim)$

(2) he_1 saw $\text{Ann}_2\text{'s}_3$ husband

he_1: np_{N}: z_1

saw: $(\text{s}\backslash\text{np}_{\text{N}})/\text{np}_{\text{A}}$: $see_{e(et)}$

$\text{Ann}_2\text{'s}_3$: $\text{np}_{\text{A}}/\text{n}$: $\lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]$

husband: n: $hsb_{e(et)}$

$>$ np_{A}: $\lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}$

$>$ $\text{s}\backslash\text{np}_{\text{N}}$: $see_{e(et)}\ \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}$

$<$ s: $see_{e(et)}\ \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}\ z_1$

4. Reducing TL-translations

D4 (TL-equivalence). TL-terms α and β are *equivalent*, written $\alpha \equiv \beta$, iff $\forall M, g$: $[\![\alpha]\!]^{M,g} \doteq [\![\beta]\!]^{M,g}$ & $[\![\beta]\!]^{M,g} \doteq [\![\alpha]\!]^{M,g}$

EXAMPLE:

- $\forall M, g$: $[\![\lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}]\!]^{M,g}$ (proof in **L4**)
  $= [\![\iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann))]\!]^{M,g}$

$\therefore$ $\lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}$
$\equiv \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann))$

D5 (free & bound variables). For any variable u and term β of TL:

- u is *bound* in β, iff u occurs in a term of the form $\lambda u[\alpha]$, or $\iota u\alpha$, contained in β.
- u is *free* in β, iff u is not bound in β.
- In $\lambda u[\alpha]$, λu *binds* the occurrence of u it contains and any free occurrence of u in α.
  In $\iota u\alpha$, ιu *binds* the occurrence of u it contains and any free occurrence of u in α.

EXAMPLES:

- in $\lambda y[\lambda x[see_{e(et)}\ yx]]\iota x(x = z_1)$ both occurrences of y are bound by λy, the 1st and 2nd occurrence of x is bound by λx, the 3rd and 4th occurrence of x is bound by ιx, z_1 is free
- in $\lambda x[see_{e(et)}\ \iota x(x = z_1)\ x]$ the 1st and 4th occurrence of x is bound by λx, the 2nd and 3rd occurrence of x is bound by ιx, z_1 is free

D6 (substitution, λ-conversion, validity). For any variable u and terms α, β of TL:

- $[\beta/u]\alpha$ is the result of substituting β for every free occurrence of u in α.
- *λ-conversion* (or *β-reduction*) converts (reduces) $\lambda u[\alpha]\beta$ into (to) $[\beta/u]\alpha$.
- λ-conversion of $\lambda u[\alpha]\beta$ into $[\beta/u]\alpha$ is *valid*, iff $\lambda u[\alpha]\beta \equiv [\beta/u]\alpha$

VALID λ-CONVERSIONS:

- $\lambda y[\lambda x[see_{e(et)}\ yx]]z$
  $\equiv \lambda x[see_{e(et)}\ zx]$ (Hwk 2a)
- $\lambda y[\lambda x[see_{e(et)}\ yx]]\iota x(x = z_1)$
  $\equiv \lambda x[see_{e(et)}\ \iota x(x = z_1)x]$
- $\lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}$
  $\equiv \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann))$

INVALID λ-CONVERSION:

- $\lambda y[\lambda x[see_{e(et)}\ yx]]x$
  $\not\equiv \lambda x[see_{e(et)}\ xx]$ (Hwk 2b)

D7 (alphabetic variants). A TL-term β is an *alphabetic variant* of a TL-term α, iff (i) β is the result of renaming one or more bound variables in α, and (ii) $\alpha \equiv \beta$.

ALPHABETIC VARIANTS:

- $\lambda y[\lambda x[see_{e(et)}\ yx]]x$
  $\equiv \lambda y[\lambda z[see_{e(et)}\ yz]]x$ (Hwk 2c)
- $\lambda x[see_{e(et)}\ \iota x(x = z_1)x]$
  $\equiv \lambda x[see_{e(et)}\ \iota y(y = z_1)x]$

NOT ALPHABETIC VARIANTS:

- $\lambda y[\lambda x[see_{e(et)}\ yx]]x$
  $\not\equiv \lambda y[\lambda x[see_{e(et)}\ yx]]z$ (Hwk 2d)

5. Semantic composition with(out) λ-conversion

A. analysis *without* λ*-conversion*:

- transparent composition
- opaque meaning

(3)

he$_1$	spoke	to	Ann$_2$'s$_3$	husband
$np_N: z_1$	$(s\backslash np_N)/np_{to}: spk_{e(et)}$	$np_{to}/np_A: \lambda x[x]$	$np_A/n: \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]$	$n: hsb_{e(et)}$

$$np_A: \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)} \quad >$$

$$np_{to}: \lambda x[x]\ \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)} \quad >$$

$$s\backslash np_N: spk_{e(et)}\ \lambda x[x]\ \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)} \quad >$$

$$s: spk_{e(et)}\ \lambda x[x]\ \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}\ z_1 \quad <$$

B. same analysis with *valid* λ*-conversion*:

- transparent composition
- transparent meaning

(3)

he$_1$	spoke	to	Ann$_2$'s$_3$	husband
$np_N: z_1$	$(s\backslash np_N)/np_{to}: spk_{e(et)}$	$np_{to}/np_A: \lambda x[x]$	$np_A/n: \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]$	$n: hsb_{e(et)}$

$$np_A: \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann)) \quad >$$

$$np_{to}: \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann)) \quad >$$

$$s\backslash np_N: spk_{e(et)}\ \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann)) \quad >$$

$$s: spk_{e(et)}\ \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann))\ z_1 \quad <$$

REFERENCES

Ajdukiewicz, K. 1935. Die syntaktische Konnexität. In S. McCall, ed. *Polish Logic 1920–1939*, 207–231. Oxford University Press, Oxford. Translated from *Studia Philosophica*, 1, 1–27.

Bar-Hillel, Y. 1953. A quasi-arithmetical notation for syntactic description. *Language* **29**:47–58.

Steedman, M. 2000. *The Syntactic Process.* MIT Press, Cambridge MA.

Lecture 4
ENGLISH FRAGMENT 1: SAMPLE ANALYSIS

1. Overview

ENGLISH SENTENCE (in Fragment 1):

(1) Ann's husband spoke to her.

ANALYSIS (in Fragment 1):

- Translation of English (1) to TL (1′) (details in Sec. **2**)

(1) $\text{Ann}_2\text{'s}_3$ husband spoke to her_2

$\text{Ann}_2\text{'s}_3$: $\text{np}_\text{N}/\text{n}$: $\lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]$

husband: n: $hsb_{e(et)}$

spoke: $(\text{s}\backslash\text{np}_\text{N})/\text{np}_{to}$: $spk_{e(et)}$

to: $\text{np}_{to}/\text{np}_\text{A}$: $\lambda x[x]$

her_2: np_A: z_2

1 ——— > np_{to}: z_2

——— > $\text{s}\backslash\text{np}_\text{N}$: $spk_{e(et)}\, z_2$

2 ——— > np_N: $\iota x(x = z_3 \wedge (hsb_{e(et)}\, z_2z_3 \wedge z_2 = ann))$

——— < s: $spk_{e(et)}\, z_2\, \iota x(x = z_3 \wedge (hsb_{e(et)}\, z_2z_3 \wedge z_2 = ann))$ =: (1′)

- Truth condition of TL (1′) (details in Sec. **3**)

$\models_M spk_{e(et)}\, z_2\, \iota x(x = z_3 \wedge (hsb_{e(et)}\, z_2z_3 \wedge z_2 = ann))$

iff $\exists d \in D_e^M$: $[\![hsb_{e(et)}]\!]^M([\![ann]\!]^M)(d) = 1$ & $[\![spk_{e(et)}]\!]^M([\![ann]\!]^M)(d) = 1$

PREDICTION:

English (1) has a reading that native speakers judge true in a situation represented by M, iff $\models_M$ (1′)

2. Proofs: λ-conversions valid

F1⁰ $[\![\lambda x[x]]\!]^{M,g} = \langle d: d \in D_e^M\rangle$ (abstract in **λ-cnv. 1)**

PROOF: (1) ≐ (4)

1. $[\![\lambda x[x]]\!]^{M,g}$
2. $\langle [\![x]\!]^{M,g[x/d]}: d \in D_e^M\rangle$ D2.3:**λ**
3. $\langle g[x/d](x): d \in D_e^M\rangle$ D2.3:**B**
4. $\langle d: d \in D_e^M\rangle$ D2.2:$g[u/d]$

□

F1¹ $[\![\lambda x[x]z_2]\!]^{M,g} = [\![z_2]\!]^{M,g}$ **(λ-cnv. 1** valid)

PROOF: (1) ≐ (4)

1. $[\![\lambda x[x]z_2]\!]^{M,g}$
2. $[\![\lambda x[x]]\!]^{M,g}([\![z_2]\!]^{M,g})$ D2.3:**A**
3. $\langle d: d \in D_e^M\rangle([\![z_2]\!]^{M,g})$ F1^0
4. $[\![z_2]\!]^{M,g}$ df. ⟨–:–⟩, D2.2

□

F2[0] For any $f \in D_{(e(et))}{}^{M}$: (abstract in **λ-cnv. 2)**

$\llbracket \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\rrbracket^{M,g}(f)$

$= \quad g(z_3)$ if $f(g(z_2))(g(z_3)) = 1$ & $g(z_2) = \llbracket ann\rrbracket^{M}$

undefined otherwise

PROOF: For any $f \in D_{(e(et))}{}^{M}$, (1) ≐ (11):

1.	$\llbracket \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\rrbracket^{M,g}(f)$	
2.	$\langle \llbracket \iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))\rrbracket^{M,g[R/d]}: d \in D_{(e(et))}{}^{M}\rangle(f)$	D2.3:**λ**
3.	$\llbracket \iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))\rrbracket^{M,g[R/f]}$	df. ⟨–:–⟩
4.	THE$\{d \in D_e{}^{M} \mid \llbracket (x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))\rrbracket^{M,g[R/f][x/d]} = 1\}$	D2.3:**ι**
5.	THE$\{d \in D_e{}^{M} \mid \llbracket (x = z_3)\rrbracket^{M,g[R/f][x/d]} = 1$	D2.3:∧, F.∩,
	& $\llbracket (Rz_2z_3 \wedge (z_2 = ann))\rrbracket^{M,g[R/f][x/d]} = 1\}$	D2.1–3
6.	THE$\{d \in D_e{}^{M} \mid \llbracket (x = z_3)\rrbracket^{M,g[R/f][x/d]} = 1$	D2.3:∧, F.∩,
	& $\llbracket Rz_2z_3\rrbracket^{M,g[R/f][x/d]} = 1$	D2.1–3
	& $\llbracket (z_2 = ann)\rrbracket^{M,g[R/f][x/d]} = 1\}$	
7.	THE$\{d \in D_e{}^{M} \mid g[R/f][x/d](x) = g[R/f][x/d](z_3)$	D2.3:=, **B**
	& $g[R/f][x/d](R)(g[R/f][x/d](z_2))(g[R/f][x/d](z_3)) = 1$	D2.3:**A**, **B**
	& $g[R/f][x/d](z_2) = \llbracket ann\rrbracket^{M}\}$	D2.3:=, **B**
8.	THE$\{d \in D_e{}^{M} \mid d = g[R/f](z_3)$	D2.2:$g[u/d]$
	& $g[R/f](R)(g[R/f](z_2))(g[R/f](z_3)) = 1$	
	& $g[R/f](z_2) = \llbracket ann\rrbracket^{M}\}$	
9.	THE$\{d \in D_e{}^{M} \mid d = g(z_3)$	D2.2:$g[u/d]$
	& $f(g(z_2))(g(z_3)) = 1$	
	& $g(z_2) = \llbracket ann\rrbracket^{M}\}$	
10.	THE$\{g(z_3)\}$ if $f(g(z_2))(g(z_3)) = 1$ & $g(z_2) = \llbracket ann\rrbracket^{M}$	df. {–:–}
	THE$\{\}$ otherwise	
11.	$g(z_3)$ if $f(g(z_2))(g(z_3)) = 1$ & $g(z_2) = \llbracket ann\rrbracket^{M}$	D2.3:THE A
	undefined otherwise	

□

F2[1] $\llbracket \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}\rrbracket^{M,g}$ (input to **λ-cnv. 2**)

$= \quad g(z_3)$ if $\llbracket hsb_{e(et)}\rrbracket^{M}(g(z_2))(g(z_3)) = 1$ & $g(z_2) = \llbracket ann\rrbracket^{M}$

undefined otherwise

PROOF: (1) ≐ (3):

1.	$\llbracket \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\ hsb_{e(et)}\rrbracket^{M,g}$	
2.	$\llbracket \lambda R[\iota x(x = z_3 \wedge (Rz_2z_3 \wedge z_2 = ann))]\rrbracket^{M,g}(\llbracket hsb_{e(et)}\rrbracket^{M})$	D2.3:**A**, **B**
3.	$g(z_3)$ if $\llbracket hsb_{e(et)}\rrbracket^{M}(g(z_2))(g(z_3)) = 1$ & $g(z_2) = \llbracket ann\rrbracket^{M}$	F2[0], D2.2
	undefined otherwise	

□

F2[2] $\llbracket \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2z_3 \wedge z_2 = ann))\rrbracket^{M,g}$ (output of **λ-cnv. 2**)

$= \quad g(z_3)$ if $\llbracket hsb_{e(et)}\rrbracket^{M}(g(z_2))(g(z_3)) = 1$ & $g(z_2) = \llbracket ann\rrbracket^{M}$

undefined otherwise

PROOF: Solution to H1

3. <u>More proofs: From TL syntax to truth condition</u>

- From TL syntax to (M, g)-denotation

F3⁰ $[\![spk_{e(et)}\ z_2\ \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2 z_3 \wedge z_2 = ann))]\!]^{M,g}$

$$= \begin{cases} [\![spk_{e(et)}]\!]^M(g(z_2))(g(z_3)) & \text{if } [\![hsb_{e(et)}]\!]^M(g(z_2))(g(z_3)) = 1 \ \&\ g(z_2) = [\![ann]\!]^M \\ \text{undefined} & \text{otherwise} \end{cases}$$

PROOF (outline): Given F2²,

$spk_{e(et)}$ ——— $(e(et))$: $[\![spk_{e(et)}]\!]^M$

z_2 ——— e: $g(z_2)$

$\iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2 z_3 \wedge z_2 = ann))$ ——— **ι**

e: $g(z_3)$ if $[\![hsb_{e(et)}]\!]^M(g(z_2))(g(z_3)) = 1$ & $g(z_2) = [\![ann]\!]^M$; undef. otherwise

——— **A**

(et): $[\![spk_{e(et)}]\!]^M(g(z_2))$

——— **A**

t: $[\![spk_{e(et)}]\!]^M(g(z_2))(g(z_3))$ if $[\![hsb_{e(et)}]\!]^M(g(z_2))(g(z_3)) = 1$ & $g(z_2) = [\![ann]\!]^M$

undefined otherwise

□

- From (M, g)-denotation to truth in M

F3¹ $\models_M spk_{e(et)}\ z_2\ \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2 z_3 \wedge z_2 = ann))$

iff $\exists d \in D_e^M$: $[\![hsb_{e(et)}]\!]^M([\![ann]\!]^M)(d) = 1$ & $[\![spk_{e(et)}]\!]^M([\![ann]\!]^M)(d) = 1$

PROOF: (1) iff (5):

1. $\models_M spk_{e(et)}\ z_2\ \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2 z_3 \wedge z_2 = ann))$
2. $\exists g$: $[\![spk_{e(et)}\ z_2\ \iota x(x = z_3 \wedge (hsb_{e(et)}\ z_2 z_3 \wedge z_2 = ann))]\!]^{M,g} = 1$ — D3:$\models_M$
3. $\exists g$:
   $[\![hsb_{e(et)}]\!]^M(g(z_2))(g(z_3)) = 1$ & $g(z_2) = [\![ann]\!]^M$ & $[\![spk_{e(et)}]\!]^M(g(z_2))(g(z_3)) = 1$ — F3⁰
4. $\exists c, d \in D_e^M$:
   $[\![hsb_{e(et)}]\!]^M(c)(d) = 1$ & $c = [\![ann]\!]^M$ & $[\![spk_{e(et)}]\!]^M(c)(d) = 1$ — D2.2
5. $\exists d \in D_e^M$:
   $[\![hsb_{e(et)}]\!]^M([\![ann]\!]^M)(d) = 1$ & $[\![spk_{e(et)}]\!]^M([\![ann]\!]^M)(d) = 1$ — elim. c

□

Homework 2

Prove the following (non-)equivalence facts:

F4 VALID λ-CONVERSION:
$\lambda y[\lambda x[see_{e(et)}\ yx]]z$
$\equiv \lambda x[see_{e(et)}\ zx]$

F5 INVALID λ-CONVERSION:
$\lambda y[\lambda x[see_{e(et)}\ yx]]x$
$\not\equiv \lambda x[see_{e(et)}\ xx]$

F6 ALPHABETIC VARIANTS:
$\lambda y[\lambda x[see_{e(et)}\ yx]]x$
$\equiv \lambda y[\lambda z[see_{e(et)}\ yz]]x$

F7 NOT ALPHABETIC VARIANTS:
$\lambda y[\lambda x[see_{e(et)}\ yx]]x$
$\not\equiv \lambda y[\lambda x[see_{e(et)}\ yx]]z$

SUGGESTED STRATEGY: Start by completing and proving the following auxiliary facts. If you start with the first two, you can use **F4^0** to prove **F4^1** and **F5^1**; and **F6^0**, to prove **F6^2**. You will also need the definition of *function abstraction*, $\langle f(a): a \in \text{Dom } f\rangle := f$ (see Appendix). Then use these facts, plus the definition of *equivalence* (D4) and function abstraction, to prove **F4** – **F7**.

F4^0 $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]]\!]^{M, g}$

= ____

F6^0 $[\![\lambda y[\lambda z[see_{e(et)}\, yz]]]\!]^{M, g}$

= ____

F4^1 $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]z]\!]^{M, g}$

= ____

F4^2 $[\![\lambda x[see_{e(et)}\, zx]]\!]^{M, g}$

= ____

F5^1 $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]x]\!]^{M, g}$

= ____

F5^2 $[\![\lambda x[see_{e(et)}\, xx]]\!]^{M, g}$

= ____

F6^2 $[\![\lambda y[\lambda z[see_{e(et)}\, yz]]x]\!]^{M, g}$

= ____

NOTES ON DEFINITION D4 (equivalence):

- To prove *equivalence*, $\alpha \equiv \beta$, you have to show that for **every** M, g, $[\![\alpha]\!]^{M, g}$ and $[\![\beta]\!]^{M, g}$ are either both undefined or both name the same semantic object (here, the same function).
- To prove *non-equivalence*, $\alpha \not\equiv \beta$, you have to show that for **some** M, g: $[\![\alpha]\!]^{M, g} \neq [\![\beta]\!]^{M, g}$. This can be done by giving an example. That is, construct a suitable model and assignment, call them M_0 and g_0, and show that $[\![\alpha]\!]^{M0, g0} \neq [\![\beta]\!]^{M0, g0}$. To show $f_1 \neq f_2$, for two functions f_1 and f_2 with the same domain A, it is enough to show that there is an argument $a \in A$ s.t. $f_1(a) \neq f_2(a)$. So you don't have to specify M_0 and g_0 in full. To keep the proof perspicuous, say just enough about M_0 and g_0 to be able to give an example of an argument a s.t. $[\![\alpha]\!]^{M0, g0}(a) \neq [\![\beta]\!]^{M0, g0}(a)$.

Solution to Homework 2

F4$^{\mathbf{0}}$ $\llbracket \lambda y[\lambda x[see_{e(et)}\ yx]]\rrbracket^{M,\,g}$
$= \langle\langle \llbracket see_{e(et)}\rrbracket^{M}(d)(c)\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$

PROOF: (1) $\doteq$ (6)

1.	$\llbracket \lambda y[\lambda x[see_{e(et)}\ yx]]\rrbracket^{M,\,g}$	
2.	$\langle \llbracket \lambda x[see_{e(et)}\ yx]\rrbracket^{M,\,g[y/d]}\colon d \in D_e^{\,M}\rangle$	D2.3:**λ**
3.	$\langle\langle \llbracket see_{e(et)}\ yx\rrbracket^{M,\,g[y/d][x/c]}\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.3:**λ**
4.	$\langle\langle \llbracket see_{e(et)}\rrbracket^{M,\,g[y/d][x/c]}(\llbracket y\rrbracket^{M,\,g[y/d][x/c]})(\llbracket x\rrbracket^{M,\,g[y/d][x/c]})\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.3:**A**
5.	$\langle\langle \llbracket see_{e(et)}\rrbracket^{M}(g[y/d][x/c](y))(g[y/d][x/c](x))\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.3:**B**
6.	$\langle\langle \llbracket see_{e(et)}\rrbracket^{M}(d)(c)\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.2:g[*u*/*d*]

□

F6$^{\mathbf{0}}$ $\llbracket \lambda y[\lambda z[see_{e(et)}\ yz]]\rrbracket^{M,\,g}$
$= \langle\langle \llbracket see_{e(et)}\rrbracket^{M}(d)(c)\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$

PROOF: (1) $\doteq$ (6)

1.	$\llbracket \lambda y[\lambda z[see_{e(et)}\ yz]]\rrbracket^{M,\,g}$	
2.	$\langle \llbracket \lambda z[see_{e(et)}\ yz]\rrbracket^{M,\,g[y/d]}\colon d \in D_e^{\,M}\rangle$	D2.3:**λ**
3.	$\langle\langle \llbracket see_{e(et)}\ yz\rrbracket^{M,\,g[y/d][z/c]}\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.3:**λ**
4.	$\langle\langle \llbracket see_{e(et)}\rrbracket^{M,\,g[y/d][z/c]}(\llbracket y\rrbracket^{M,\,g[y/d][z/c]})(\llbracket z\rrbracket^{M,\,g[y/d][z/c]})\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.3:**A**
5.	$\langle\langle \llbracket see_{e(et)}\rrbracket^{M}(g[y/d][z/c](y))(g[y/d][z/c](z))\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.3:**B**
6.	$\langle\langle \llbracket see_{e(et)}\rrbracket^{M}(d)(c)\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle$	D2.2:g[*u*/*d*]

□

F4$^{\mathbf{1}}$ $\llbracket \lambda y[\lambda x[see_{e(et)}\ yx]]z\rrbracket^{M,\,g}$
$= \langle \llbracket see_{e(et)}\rrbracket^{M}(g(z))(c)\colon c \in D_e^{\,M}\rangle$

PROOF: (1) $\doteq$ (5)

1.	$\llbracket \lambda y[\lambda x[see_{e(et)}\ yx]]z\rrbracket^{M,\,g}$	
2.	$\llbracket \lambda y[\lambda x[see_{e(et)}\ yx]]\rrbracket^{M,\,g}(\llbracket z\rrbracket^{M,\,g})$	D2.3:**A**
3.	$\llbracket \lambda y[\lambda x[see_{e(et)}\ yx]]\rrbracket^{M,\,g}(g(z))$	D2.3:**B**
4.	$\langle\langle \llbracket see_{e(et)}\rrbracket^{M}(d)(c)\colon c \in D_e^{\,M}\rangle\colon d \in D_e^{\,M}\rangle(g(z))$	$\mathrm{F4}^0$
5.	$\langle \llbracket see_{e(et)}\rrbracket^{M}(g(z))(c)\colon c \in D_e^{\,M}\rangle$	D2.2, df. $\langle$–:–$\rangle$

□

F4$^{\mathbf{2}}$ $\llbracket \lambda x[see_{e(et)}\ zx]\rrbracket^{M,\,g}$
$= \langle \llbracket see_{e(et)}\rrbracket^{M}(g(z))(c)\colon c \in D_e^{\,M}\rangle$

PROOF: (1) $\doteq$ (5)

1.	$\llbracket \lambda x[see_{e(et)}\ zx]\rrbracket^{M,\,g}$	
2.	$\langle \llbracket see_{e(et)}\ zx\rrbracket^{M,\,g[x/c]}\colon c \in D_e^{\,M}\rangle$	D2.3:**λ**
3.	$\langle \llbracket see_{e(et)}\rrbracket^{M,\,g[x/c]}(\llbracket z\rrbracket^{M,\,g[x/c]})(\llbracket x\rrbracket^{M,\,g[x/c]})\colon c \in D_e^{\,M}\rangle$	D2.3:**A**
4.	$\langle \llbracket see_{e(et)}\rrbracket^{M}(g[x/c](z))(g[x/c](x))\colon c \in D_e^{\,M}\rangle$	D2.3:**B**
5.	$\langle \llbracket see_{e(et)}\rrbracket^{M}(g(z))(c)\colon c \in D_e^{\,M}\rangle$	D2.2:g[*u*/*d*]

□

F5[1] $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]x]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^M(g(x))(c){:}\ c \in D_e{}^M \rangle$

PROOF: (1) ≐ (5)
1. $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]x]\!]^{M,g}$
2. $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]]\!]^{M,g}([\![x]\!]^{M,g})$ D2.3:**A**
3. $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]]\!]^{M,g}(g(x))$ D2.3:**B**
4. $\langle\langle [\![see_{e(et)}]\!]^M(d)(c){:}\ c \in D_e{}^M\rangle{:}\ d \in D_e{}^M\rangle(g(x))$ $F4^0$
5. $\langle [\![see_{e(et)}]\!]^M(g(x))(c){:}\ c \in D_e{}^M\rangle$ D2.2, df. ⟨–:–⟩

□

F5[2] $[\![\lambda x[see_{e(et)}\, xx]]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^M(c)(c){:}\ c \in D_e{}^M \rangle$

PROOF: (1) ≐ (5)
1. $[\![\lambda x[see_{e(et)}\, xx]]\!]^{M,g}$
2. $\langle [\![see_{e(et)}\, xx]\!]^{M,g[x/c]}{:}\ c \in D_e{}^M\rangle$ D2.3:**λ**
3. $\langle [\![see_{e(et)}]\!]^{M,g[x/c]}([\![x]\!]^{M,g[x/c]})([\![x]\!]^{M,g[x/c]}){:}\ c \in D_e{}^M\rangle$ D2.3:**A**
4. $\langle [\![see_{e(et)}]\!]^M(g[x/c](x))(g[x/c](x)){:}\ c \in D_e{}^M\rangle$ D2.3:**B**
5. $\langle [\![see_{e(et)}]\!]^M(c)(c){:}\ c \in D_e{}^M\rangle$ D2.2:$g[u/d]$

□

F6[2] $[\![\lambda y[\lambda z[see_{e(et)}\, yz]]x]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^M(g(x))(c){:}\ c \in D_e{}^M \rangle$

PROOF: (1) ≐ (5)
1. $[\![\lambda y[\lambda z[see_{e(et)}\, yz]]x]\!]^{M,g}$
2. $[\![\lambda y[\lambda z[see_{e(et)}\, yz]]]\!]^{M,g}([\![x]\!]^{M,g})$ D2.3:**A**
3. $[\![\lambda y[\lambda z[see_{e(et)}\, yz]]]\!]^{M,g}(g(x))$ D2.3:**B**
4. $\langle\langle [\![see_{e(et)}]\!]^M(d)(c){:}\ c \in D_e{}^M\rangle{:}\ d \in D_e{}^M\rangle(g(x))$ $F6^0$
5. $\langle [\![see_{e(et)}]\!]^M(g(x))(c){:}\ c \in D_e{}^M\rangle$ D2.2, df. ⟨–:–⟩

□

F4 VALID λ-CONVERSION:
$\lambda y[\lambda x[see_{e(et)}\, yx]]z$
$\equiv \lambda x[see_{e(et)}\, zx]$

PROOF: For all M, g:
$[\![\lambda y[\lambda x[see_{e(et)}\, yx]]z]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^M(g(z))(c){:}\ c \in D_e{}^M\rangle$ $F4^1$
$= [\![\lambda x[see_{e(et)}\, zx]]\!]^{M,g}$ $F4^2$
Hence, by D4, $\lambda y[\lambda x[see_{e(et)}\, yx]]z \equiv \lambda x[see_{e(et)}\, zx]$

□

F5 INVALID λ-CONVERSION:

$\lambda y[\lambda x[see_{e(et)}\, yx]]x$
$\not\equiv \lambda x[see_{e(et)}\, xx]$

PROOF: Consider M_0, g_0, s.t. $\{J, A\} \subseteq D_e^{M0}$ and (i)–(iii):

(i) $[\![see_{e(et)}]\!]^{M0}(J)(A) = 1$ (ii) $[\![see_{e(et)}]\!]^{M0}(A)(A) = 0$ (iii) $g_0(x) = J$

Then:

- $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]x]\!]^{M0, g0}(A)$

=	$\langle [\![see_{e(et)}]\!]^{M0}(g_0(x))(c): c \in D_e^{M0}\rangle(A)$	F5[1]
=	$[\![see_{e(et)}]\!]^{M0}(g_0(x))(A)$	D_e^{M0}, $\langle - : - \rangle$
=	1	(iii), (i)

- $[\![\lambda x[see_{e(et)}\, xx]]\!]^{M0, g0}(A)$

=	$\langle [\![see_{e(et)}]\!]^{M0}(c)(c): c \in D_e^{M0}\rangle(A)$	F5[2]
=	$[\![see_{e(et)}]\!]^{M0}(A)(A)$	D_e^{M0}, $\langle - : - \rangle$
=	0	(ii)

Hence, by D4, $\lambda y[\lambda x[see_{e(et)}\, yx]]x \not\equiv \lambda x[see_{e(et)}\, xx]$

□

<u>Accidental binding</u>:

The λ-conversion in **F5** is invalid because of ACCIDENTAL BINDING. That is, there is an occurrence of a variable that is *free in the input* (the last occurrence of x),

$$\lambda y[\lambda x[see_{e(et)}\, yx]]\boldsymbol{x}$$

whereas the corresponding variable occurrence is *bound in the output*:

$$\lambda x[see_{e(et)}\, \boldsymbol{x}x]$$

Therefore, the denotation of the input depends on the assignment to that variable,

$$[\![\lambda y[\lambda x[see_{e(et)}\, yx]]\boldsymbol{x}]\!]^{M, g}$$
$$= \langle [\![see_{e(et)}]\!]^{M}(g(\boldsymbol{x}))(c): c \in D_e^{M}\rangle \quad \text{F5}^1$$

whereas the denotation of the output does not:

$$[\![\lambda x[see_{e(et)}\, \boldsymbol{x}x]]\!]^{M, g}$$
$$= \langle [\![see_{e(et)}]\!]^{M}(c)(c): c \in D_e^{M}\rangle \quad \text{F5}^2$$

In general, failure to preserve variable-binding relations leads to failure to preserve meaning. Therefore, λ-conversions that accidentally bind free variables (by bringing them into the scope of a potential binder, here λx) are *invalid*.

F6 ALPHABETIC VARIANTS:
$\lambda y[\lambda x[see_{e(et)}\, yx]]x$
$\equiv \lambda y[\lambda z[see_{e(et)}\, yz]]x$

PROOF: For all M, g:
$[\![\lambda y[\lambda x[see_{e(et)}\, yx]]x]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^{M}(g(x))(c){:}\ c \in D_e^{M}\rangle$ F4[1]
$= [\![\lambda y[\lambda z[see_{e(et)}\, yz]]x]\!]^{M,g}$ F6[2]
Hence, by D4, $\lambda y[\lambda x[see_{e(et)}\, yx]]x \equiv \lambda y[\lambda z[see_{e(et)}\, yz]]x$

□

F7 NOT ALPHABETIC VARIANTS:
$\lambda y[\lambda x[see_{e(et)}\, yx]]x$
$\not\equiv \lambda y[\lambda x[see_{e(et)}\, yx]]z$

PROOF: Consider M_0, g_0, s.t. $\{J, A\} \subseteq D_e^{M0}$ and (i)–(iv):
(i) $[\![see_{e(et)}]\!]^{M0}(J)(A) = 1$ (ii) $[\![see_{e(et)}]\!]^{M0}(A)(A) = 0$ (iii) $g_0(x) = J$ (iv) $g_0(z) = A$
Then:
- $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]x]\!]^{M0,\, g0}(A)$
$= \langle [\![see_{e(et)}]\!]^{M0}(g_0(x))(c){:}\ c \in D_e^{M0}\rangle(A)$ F5[1]
$= [\![see_{e(et)}]\!]^{M0}(g_0(x))(A)$ D_e^{M0}, $\langle - {:} - \rangle$
$= 1$ (iii), (i)
- $[\![\lambda y[\lambda x[see_{e(et)}\, yx]]z]\!]^{M0,\, g0}(A)$
$= \langle [\![see_{e(et)}]\!]^{M0}(g_0(z))(c){:}\ c \in D_e^{M0}\rangle(A)$ F4[1]
$= [\![see_{e(et)}]\!]^{M0}(g_0(z))(A)$ D_e^{M0}, $\langle - {:} - \rangle$
$= 0$ (iv), (ii)

Hence, by D4, $\lambda y[\lambda x[see_{e(et)}\, yx]]x \not\equiv \lambda y[\lambda x[see_{e(et)}\, yx]]z$

□

<u>Renaming variables</u>:
- OK to rename a *bound* variable, because this has no effect on the denotation, e.g.

$[\![\lambda y[\lambda \boldsymbol{x}[see_{e(et)}\, y\boldsymbol{x}]]x]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^{M}(g(x))(\boldsymbol{c}){:}\ \boldsymbol{c} \in D_e^{M}\rangle$ F4[1]
$= [\![\lambda y[\lambda \boldsymbol{z}[see_{e(et)}\, y\boldsymbol{z}]]x]\!]^{M,g}$ F6[2]

- DO NOT rename a *free* variable, because the denotation depends on the values assigned to free variables, so changing a free variable may change the denotation (see below and proof of **F7**):

$[\![\lambda y[\lambda x[see_{e(et)}\, yx]]\boldsymbol{x}]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^{M}(g(\boldsymbol{x}))(c){:}\ c \in D_e^{M}\rangle$ F5[1]

$[\![\lambda y[\lambda x[see_{e(et)}\, yx]]\boldsymbol{z}]\!]^{M,g}$
$= \langle [\![see_{e(et)}]\!]^{M}(g(\boldsymbol{z}))(c){:}\ c \in D_e^{M}\rangle$ F4[1]

- BOTTOM LINE: Meaning-preserving operations preserve (i) variable-binding relations, and (ii) don't change any free variables.