Alex Silk
Heim & Kratzer "Cheat Sheet"(ish)

Sets, relations, functions jargon

- $\{o : o = \text{Penny}\}$ = the set of things o such that o is identical to Penny = $\{\text{Penny}\}$
- relation: a set of ordered pairs $\langle a, b\rangle$ (where $\langle a, b\rangle$ is the pair of elements a and b in that order)
- A function $f : X \to Y$ is a relation that associates any element in X (the domain of f) with a unique element in Y (the range of f) — formally, a set S of ordered pairs such that if $\langle a, b\rangle \in S$ and $\langle a, c\rangle \in S$, then $b = c$.
  - e.g., the relation $f_i = \{\langle 1, \text{Alice}\rangle, \langle 2, \text{Bert}\rangle, \langle 3, \text{Alice}\rangle\}$ is a function; the relation $R_i = \{\langle 1, 1\rangle, \langle 1, 2\rangle\}$ isn't a function.
  - domain of a function: the set of (possible) inputs to the function (i.e., first members of the pairs). the domain of $f_i = \{1, 2, 3\}$
  - range of a function: the set of (possible) outputs of the function (i.e., second members of the pairs). the range of $f_i = \{\text{Alice, Bert}\}$
- λ notation (pp. 36–37): Read "$[\lambda\alpha : \phi \,.\, \gamma]$" as
  - (a) "the function which maps every α such that ϕ to 1 iff γ" if γ is a sentence,
    - Read "$[\lambda x : x = \text{Percy or } x = \text{Colin} \,.\, x \text{ jumped}]$" as "the function which maps every x such that x = Percy or x = Colin to 1 iff x jumped"
  - (b) "the function which maps every α such that ϕ to γ" otherwise
    - Read "$[\lambda x : x = \text{Percy or } x = \text{Colin} \,.\, \text{Penny}]$" as "the function which maps every x such that x = Percy or x = Colin to Penny"

Components of our semantic theory

- *Lexicon*
  - dictionary specifying the conventional meanings of words ("lexical items")
  - includes "lexical entries" — e.g. $[\![\textbf{Alice}]\!]$ = Alice; $[\![\textbf{jumped}]\!] = [\lambda x_e \,.\, x \text{ jumped}]$
- *Interpretation rules*: rules specifying how to determine the meanings of (complex) expressions from the meanings of their parts (e.g., Function Application; see below)
- $[\![\ ]\!]$: the *interpretation function*, which maps well-formed expressions — i.e., *trees*, or words with a particular structure, as determined by the syntax — to their meanings ("denotations")
  - in our theory, the meanings of expressions are treated as elements in a "domain" of things *in the world*…
- *Semantic type*: a kind of thing. *Domain*: a set of things of a given semantic type. (p. 28)
  - $D_e = D$ = the set of things of type e = the set of individuals (say, $\{\text{Alice, Bert, …}\}$)
  - D_t = the set of things of type t = the set of truth values = $\{0, 1\}$ (where 0 represents falsity and 1 represents truth)

- $D_{\langle a,b\rangle}$ = the set of things of type $\langle a, b\rangle$ = the set of functions mapping elements in D_a to elements in D_b
  - defined as follows: (i) e and t are types; (ii) if σ and τ are types, then $\langle \sigma, \tau\rangle$ is a type; (iii) nothing else is a type.
  - e.g., $D_{\langle e,t\rangle}$ is the set of functions from individuals (type e) to truth values (type t)
  - e.g., $D_{\langle\langle e,t\rangle,e\rangle}$ is the set of functions from elements of $D_{\langle e,t\rangle}$ (type $\langle e, t\rangle$, as defined above) to individuals (type e)
- (NB: we'll often abbreviate lambda expressions of the form "$[\lambda\alpha : \alpha \in D_\tau . \gamma]$" as "$[\lambda\alpha_\tau . \gamma]$"; more generally, we'll abbreviate "$[\lambda\alpha : \psi \wedge \alpha \in D_\tau . \gamma]$" as "$[\lambda\alpha_\tau : \psi . \gamma]$" — i.e., indicating constraints that the variable α is of such-and-such semantic type τ via the subscripted α_τ.)

Inventory of justifications in derivations (details on next page):

- *Interpretation rules*
  - Function Application (FA): p. 44, 95
  - Predicate Modification (PM): p. 65, 95
  - Pronouns & Traces (PT): p. 111
  - Predicate Abstraction (PA): p. 114, 186
  - Terminal Nodes / Lexical entry ("by lexical entry for…"): p. 95
  - Semantic vacuity ("by vacuity of 'a'/'is'/'to'/'that' "): p. 62
- *Formal conventions*
  - Convention (9): p. 94
  - Convention (11): p. 112
- *Math-ish*
  - λ-reduction
  - Assignment modification: p. 112

Details:

- *Interpretation rules*
  - Function Application (FA): p. 44, 95
    - ≈ If $\alpha = \beta\gamma$ and the domain of $[\![\beta]\!]^g$ contains $[\![\gamma]\!]^g$, then $[\![\alpha]\!]^g = [\![\beta]\!]^g([\![\gamma]\!]^g)$
    - e.g., $[\![\textbf{Percy laughed}]\!]^g = [\![\textbf{laughed}]\!]^g([\![\textbf{Percy}]\!]^g)$
  - Predicate Modification (PM): p. 65, 95
    - ≈ If $\alpha = \beta\gamma$ and $[\![\beta]\!]^g, [\![\gamma]\!]^g \in D_{\langle e,t\rangle}$, then $[\![\alpha]\!]^g = [\lambda x_e \,.\, [\![\beta]\!]^g(x) = 1 \text{ and } [\![\gamma]\!]^g(x) = 1]$
    - e.g., $[\![\textbf{green pig}]\!]^g = [\lambda x_e \,.\, [\![\textbf{green}]\!]^g(x) = 1 \text{ and } [\![\textbf{pig}]\!]^g(x) = 1]$
  - Pronouns & Traces (PT): p. 111
    - ≈ $[\![\alpha_n]\!]^g = g(n)$
    - e.g., $[\![\textbf{it}_4]\!]^g = g(4)$
    - e.g., $[\![\textbf{t}_8]\!]^g = g(8)$
  - Predicate Abstraction (PA): p. 114, 186
    - ≈ If $\alpha = n\ \gamma$, then $[\![\alpha]\!]^g = [\lambda x_e \,.\, [\![\gamma]\!]^{g^{x/n}}]$
    - e.g., $[\![\textbf{which}_3\ \textbf{t}_3\ \textbf{pig}]\!]^g = [\lambda x_e \,.\, [\![\textbf{t}_3\ \textbf{pig}]\!]^{g^{x/3}}]$
  - Terminal Nodes / Lexical entry ("by lexical entry for…"): p. 95
    - e.g., $[\![\textbf{pig}]\!] = [\lambda x_e \,.\, x \text{ is a pig}]$
  - Semantic vacuity ("by vacuity of 'a'/'is'/'to'/'that' "): p. 62
    - e.g., $[\![\textbf{a pig}]\!]^g = [\![\textbf{pig}]\!]^g$
- *Formal conventions*
  - Convention (9): p. 94
    - e.g., $[\![\textbf{pig}]\!]^g = [\![\textbf{pig}]\!]$
  - Convention (11): p. 112
    - e.g., $[\![\textbf{which}_1\ \textbf{t}_1\ \textbf{pig}]\!] = [\![\textbf{which}_1\ \textbf{t}_1\ \textbf{pig}]\!]^{\varnothing}$
- *Math-ish*
  - λ-reduction
    - e.g., $[\lambda x_e \,.\, [\lambda y_e \,.\, y \text{ likes } x]](\text{Percy})(\text{Penny}) = [\lambda y_e \,.\, y \text{ likes Percy}](\text{Penny})$
  - Assignment modification: p. 112
    - "$g^{x/n}$" ≈ "the unique assignment which maps n to x and is otherwise identical to g"
    - e.g., $[\![\textbf{t}_3\ \textbf{pig}]\!]^{\varnothing^{y/3}} = [\![\textbf{t}_3\ \textbf{pig}]\!]^{[3\to y]}$

Presupposition

- *Formal implementation:*
  - **undefined** semantic values: constraints on function domains, i.e. **partial functions**
  - $[\![\alpha]\!] = [\lambda h\!:\!\underline{\quad}\,.\cdots]$
  - e.g., $[\![\textbf{the}]\!] = [\lambda f\!:\!\underline{f \in D_{\langle e,t\rangle} \text{ and there is a unique } x \text{ such that } f(x) = 1}\,.$ the unique y such that $f(y) = 1]$
  - e.g., $[\![\textbf{The cat is happy}]\!] = 1$ iff the unique y such that y is a cat is happy, provided that there is a unique x such that x is a cat
- *Diagnostics:*
  - *"Hey, Wait a Minute" test*
    - Presupposed implications, unlike truth-conditional implications, often can only be **challenged *indirectly***
    - e.g., if I said 'The Waitrose in Edgbaston is exciting', and you wanted to challenge my implication that there is a (unique) Waitrose in Edgbaston, you would need to say something like 'Hey, wait a minute; I had no idea there was a Waitrose in Edgbaston' rather than simply 'No'.
    - Contrast: if I said 'Waitrose is exciting', and you wanted to challenge my implication that Waitrose is exciting, it would be weird to say 'Hey, wait a minute; I had no idea Waitrose was exciting'. You would just say 'No, Waitrose isn't exciting'.
  - *"Family of Sentences" test*
    - Presupposed implications, unlike truth-conditional implications, are typically still taken to be **commitments** of the speaker even when the sentence is **embedded under clausal negation, in a question, or in a supposition**.
    - 'The Waitrose in Edgbaston is exciting' ⇒ there is a unique Waitrose in Edgbaston

      'It's not the case that the Waitrose in Edgbaston is exciting' ⇒ there is a unique Waitrose in Edgbaston

      'Is the Waitrose in Edgbaston exciting?' ⇒ there is a unique Waitrose in Edgbaston

      'Maybe the Waitrose in Edbaston is exciting' ⇒ there is a unique Waitrose in Edgbaston
    - 'Waitrose is exciting' ⇒ Waitrose is exciting

      'It's not the case that Waitrose is exciting' ⇏ Waitrose is exciting

      'Is Waitrose exciting?' ⇏ Waitrose is exciting

      'Maybe Waitrose is exciting' ⇏ Waitrose is exciting
  - Satisfying these tests is taken as evidence that an implication is linguistically *presupposed*.

- E.g., the implication of 'The Waitrose in Edgbaston is exciting' that there is a unique Waitrose in Edgbaston would be treated as *presupposed*. It's a condition that must be accepted in order for the sentence to be felicitously used, or for one to be in a position to evaluate it for truth or falsity.
- Contrast: the implication of 'Waitrose is exciting' that Waitrose is exciting is treated as part of the sentence's truth conditions.

Alex Silk
Heim & Kratzer, Chapter 1, pp. 9–10

a. true iff $a = b$

b. true

c. true

d. true iff anyone who likes a likes b and anyone who likes b likes a (special case: $a = b$)

e. true

f. true

g. true iff either (i) no one likes anyone, or (ii) someone likes herself and everyone is liked by someone or other

- The Left Set (LS) is $\{x\colon \{y\colon y \text{ likes } x\} = \varnothing\}$. That is, it is the set of individuals x such that the set of individuals y who like them is empty: $\{x\colon \text{no one likes } x\}$, the set of people whom no one likes.
- The Right Set (RS) is a bit trickier: The inner condition $\{x\colon x \text{ likes } x\} = \varnothing$ is the condition that no one likes themselves. But note that this condition is *independent* of the outer variable x; the outer variable x doesn't "bind" anything. We could have specified RS equivalently with $\{x\colon \{y\colon y \text{ likes } y\} = \varnothing\}$.

  What happens when the condition is independent of the outer variable? The set is either the set of all individuals D (if the condition is true), or the empty set $\varnothing$ (if the proposition is false). (Compare: (a) What is $\{z\colon 2 + 2 = 4\}$? To be in the set, you have to be such that 2+2=4. But since 2+2=4, *every* individual z is such that 2+2=4. (b) What is $\{v\colon 2 + 2 = 5\}$? To be in the set, you have to be such that 2+2=5. But 2+2 doesn't equal 5. So, no v is such that 2+2=5.)

  So, RS = D if the inner condition of the RS is true, i.e. if no one likes themselves. And RS = $\varnothing$ if the inner condition is false, i.e. if someone likes herself.
- So, returning to the original question, when is LS = RS? When either (i) LS = RS = D, or (ii) LS = RS = $\varnothing$. Take each case in turn:

  (i) LS = D. So, everyone is such that no one likes them, i.e. no one is liked by anyone. If RS also equals D, then the inner condition of RS must be true, i.e. no one likes themselves. So, LS = RS = D when no one likes anyone.

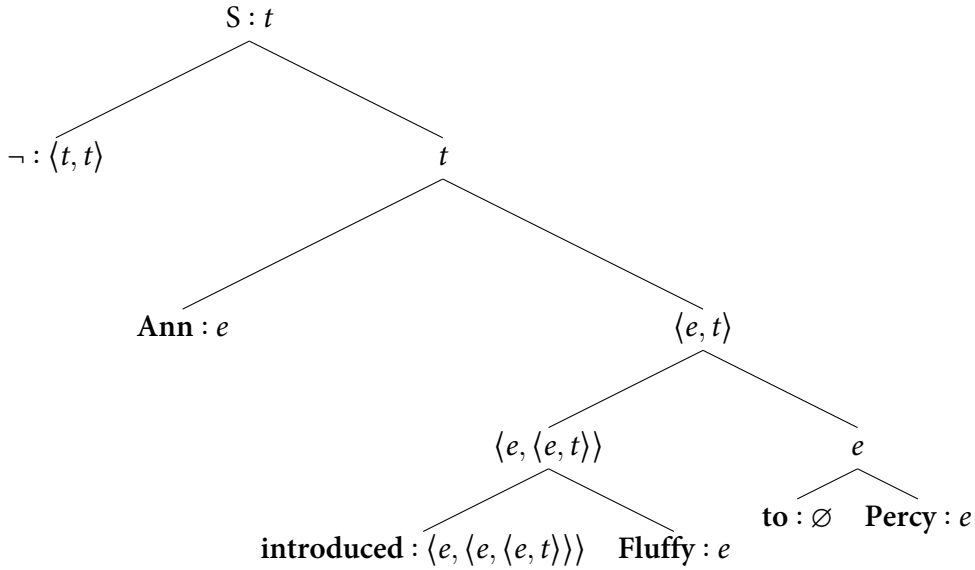
  (ii) LS = $\varnothing$. So, the set of individuals whom no one likes is empty, i.e. everyone is liked by someone or other. If RS also equals $\varnothing$, then the inner condition of RS must be false, i.e. someone likes herself. So, LS = RS = $\varnothing$ when everyone is liked by someone or other, and someone likes herself.

  So, LS = RS just when (i) no one likes anyone, or (ii) everyone is liked by someone and someone likes herself.

Alex Silk
Heim & Kratzer, Chapter 2–3ish sample derivation

It's-not-the-case-that Ann introduced Fluffy to Percy.

- (NB: ¬ = **it's-not-the-case-that**)



Lexical entries:
⟦**Ann**⟧ = Ann
⟦**Fluffy**⟧ = Fluffy
⟦**Percy**⟧ = Percy
⟦**introduced**⟧ = [λx_e.[λy_e.[λz_e . z introduced x to y]]]
⟦¬⟧ = [λp_t . p = 0]

⟦**S**⟧
= (by FA)
⟦¬⟧(⟦**Ann introduced Fluffy to Percy**⟧)
= (by FA)
⟦¬⟧(⟦**introduced Fluffy to Percy**⟧(⟦**Ann**⟧))
= (by FA)
⟦¬⟧(⟦**introduced Fluffy**⟧(⟦**to Percy**⟧)(⟦**Ann**⟧))
= (by FA)
⟦¬⟧(⟦**introduced**⟧(⟦**Fluffy**⟧)(⟦**to Percy**⟧)(⟦**Ann**⟧))
= (by vacuity of **to**)
⟦¬⟧(⟦**introduced**⟧(⟦**Fluffy**⟧)(⟦**Percy**⟧)(⟦**Ann**⟧))
= (by lexical entry for **Fluffy**)
⟦¬⟧(⟦**introduced**⟧(Fluffy)(⟦**Percy**⟧)(⟦**Ann**⟧))
= (by lexical entry for **Percy**)
⟦¬⟧(⟦**introduced**⟧(Fluffy)(Percy)(⟦**Ann**⟧))
= (by lexical entry for **Ann**)
⟦¬⟧(⟦**introduced**⟧(Fluffy)(Percy)(Ann))
= (by lexical entry for **introduced**)
⟦¬⟧([λx_e.[λy_e.[λz_e . z introduced x to y]]](Fluffy)(Percy)(Ann))

= (by λ-reduction)

$[\![\neg]\!]([\lambda y_e.[\lambda z_e\,.\,z$ introduced Fluffy to $y]](\text{Percy})(\text{Ann}))$

= (by λ-reduction)

$[\![\neg]\!]([\lambda z_e\,.\,z$ introduced Fluffy to Percy$](\text{Ann}))$

= (by lexical entry for $\neg$)

$[\lambda p_t\,.\,p = 0]([\lambda z_e\,.\,z$ introduced Fluffy to Percy$](\text{Ann}))$

= (by λ-reduction)

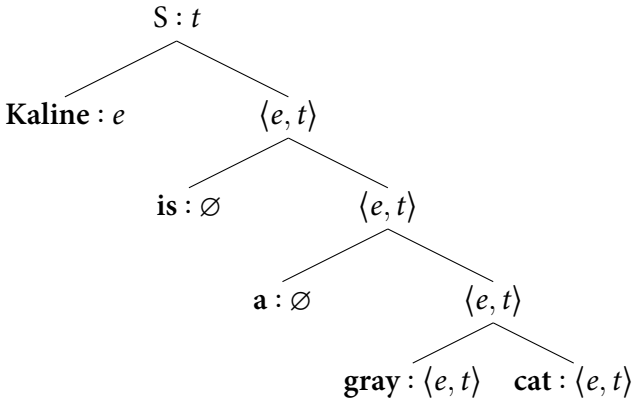
1 iff $[\lambda z_e\,.\,z$ introduced Fluffy to Percy$](\text{Ann}) = 0$

iff (by λ-reduction)

it's not the case that Ann introduced Fluffy to Percy

Alex Silk
Heim & Kratzer, Chapter 4 selected exercises

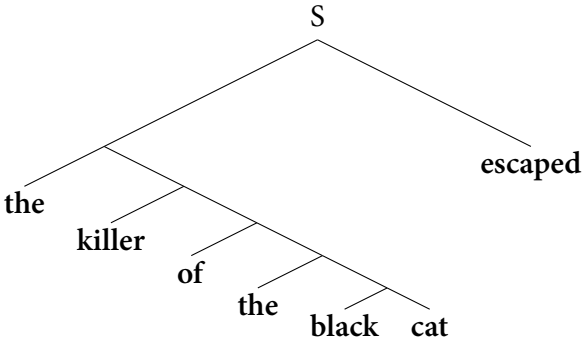
(1) Chapter 4, p. 63
Kaline is a gray cat.



⟦**S**⟧
= (by FA)
⟦**is a gray cat**⟧(⟦**Kaline**⟧)
= (by semantic vacuity of **is**)
⟦**a gray cat**⟧(⟦**Kaline**⟧)
= (by semantic vacuity of **a**)
⟦**gray cat**⟧(⟦**Kaline**⟧)
= (by lexical entry for **Kaline**)
⟦**gray cat**⟧(Kaline)
= (by PM)
$[\lambda x_e \,.\, [\![\textbf{gray}]\!](x) = [\![\textbf{cat}]\!](x) = 1](\text{Kaline})$
= (by lexical entry for **gray**)
$[\lambda x_e \,.\, [\lambda y_e \,.\, y \text{ is gray}](x) = [\![\textbf{cat}]\!](x) = 1](\text{Kaline})$
= (by lexical entry for **cat**)
$[\lambda x_e \,.\, [\lambda y_e \,.\, y \text{ is gray}](x) = [\lambda z_e \,.\, z \text{ is a cat}](x) = 1](\text{Kaline})$
= (by λ-reduction)
$[\lambda x_e \,.\, x \text{ is gray and } x \text{ is a cat}](\text{Kaline})$
= (by λ-reduction)
1 iff Kaline is gray and Kaline is a cat

(2) Chapter 4, p. 76

a. **The killer of the black cat escaped.**



b. (i) There's a unique killer of a unique black cat, but that killer didn't escape.
(ii) It's not the case that there's a unique black cat.
(iii) There's a unique black cat, but it's not the case that there's a unique killer.

(3) Chapter 4, Exercise 2, p. 80

a. (i) b_1
(ii) b_2
(iii) there is no apple that fits that description

b. $[\![\textbf{leftmost}]\!] = [\lambda x_e \,.\, x \text{ is leftmost}]$

$[\![\textbf{the leftmost dark apple-in-the-row}]\!]$
= (by FA)
$[\![\textbf{the}]\!]([\![\textbf{leftmost dark apple-in-the-row}]\!])$
= (by PM)
$[\![\textbf{the}]\!]([\lambda x_e \,.\, [\![\textbf{leftmost}]\!](x) = [\![\textbf{dark apple-in-the-row}]\!](x) = 1])$
= (by lexical entry for **leftmost**)
$[\![\textbf{the}]\!]([\lambda x_e \,.\, [\lambda y_e \,.\, y \text{ is leftmost}](x) = [\![\textbf{dark apple-in-the-row}]\!](x) = 1])$
= (by PM)
$[\![\textbf{the}]\!]([\lambda x_e \,.\, [\lambda y_e \,.\, y \text{ is leftmost}](x) = [\lambda z_e \,.\, [\![\textbf{dark}]\!](z) = [\![\textbf{apple-in-the-row}]\!](z) = 1](x) = 1])$
= (by lexical entries for **dark, apple-in-the-row**)
$[\![\textbf{the}]\!]([\lambda x_e \,.\, [\lambda y_e \,.\, y \text{ is leftmost}](x) = [\lambda z_e \,.\, [\lambda u_e \,.\, u \text{ is dark}](z) =$
$[\lambda v \,.\, v \text{ is an apple-in-the-row}](z) = 1](x) = 1])$
= (by λ-reduction)
$[\![\textbf{the}]\!]([\lambda x_e \,.\, [\lambda y_e \,.\, y \text{ is leftmost}](x) = [\lambda z_e \,.\, z \text{ is dark and } z \text{ is an apple-in-the-row}](x) = 1])$
= (by λ-reduction)
$[\![\textbf{the}]\!]([\lambda x_e \,.\, x \text{ is leftmost and } x \text{ is dark and } x \text{ is an apple-in-the-row}])$
= (by lexical entry for **the**)
$[\lambda f_{\langle e,t\rangle} : \text{there is exactly one } y \text{ such that } f(y) = 1 \,.\, \text{the unique } z \text{ such that } f(z) = 1]$
$([\lambda x_e \,.\, x \text{ is leftmost and } x \text{ is dark and } x \text{ is an apple-in-the-row}])$
= (by λ-reduction)
the unique z such that $[\lambda x_e \,.\, x \text{ is leftmost and } x \text{ is dark and } x \text{ is an apple-in-the-row}](z) = 1$,
provided that there is exactly one y such that $[\lambda x_e \,.\, x \text{ is leftmost and } x \text{ is dark and } x \text{ is}$
$\text{an apple-in-the-row}](y) = 1$

= (by λ-reduction)
the unique z such that z is leftmost and z is dark and z is an apple-in-the-row,
provided that there is exactly one y such that y is leftmost and y is dark and y is an apple-in-the-row

This says that **the leftmost dark apple in the row** picks out the unique apple in the row that is leftmost and dark. So, treating **leftmost** as a one-place predicate predicts that (ii), like (iii), won't yield a semantic value. But (ii) *does* yield a semantic value. So treating **leftmost** as a one-place predicate as above can't be right.

c. $[\![\textbf{leftmost}]\!] = [\lambda f_{\langle e,t\rangle} . [\lambda x_e . f(x) = 1 \text{ and } x \text{ is the leftmost element of } \{y : f(y) = 1\}]]$

d. $[\![\textbf{the leftmost dark apple-in-the-row}]\!]$
= (by FA)
$[\![\textbf{the}]\!]([\![\textbf{leftmost dark apple-in-the-row}]\!])$
= (by FA)
$[\![\textbf{the}]\!]([\![\textbf{leftmost}]\!]([\![\textbf{dark apple-in-the-row}]\!]))$
= (by PM)
$[\![\textbf{the}]\!]([\![\textbf{leftmost}]\!]([\lambda x_e . [\![\textbf{dark}]\!](x) = [\![\textbf{apple-in-the-row}]\!](x) = 1]))$
= (by lexical entries for **dark, apple-in-the-row**)
$[\![\textbf{the}]\!]([\![\textbf{leftmost}]\!]([\lambda x_e . [\lambda y_e . y \text{ is dark}](x) = [\lambda z_e . z \text{ is an apple-in-the-row}](x) = 1]))$
= (by λ-reduction)
$[\![\textbf{the}]\!]([\![\textbf{leftmost}]\!]([\lambda x_e . x \text{ is dark and } x \text{ is an apple-in-the-row}]))$
= (by lexical entry for **leftmost**)
$[\![\textbf{the}]\!]([\lambda f_{\langle e,t\rangle} . [\lambda y_e . f(y) = 1 \text{ and } y \text{ is the leftmost element of } \{z : f(z) = 1\}]]$
$([\lambda x_e . x \text{ is dark and } x \text{ is an apple-in-the-row}]))$
= (by λ-reduction)
$[\![\textbf{the}]\!]([\lambda y_e . [\lambda x_e . x \text{ is dark and } x \text{ is an apple-in-the-row}](y) = 1 \text{ and } y \text{ is the leftmost}$
element of $\{z : [\lambda x_e . x \text{ is dark and } x \text{ is an apple-in-the-row}](z) = 1\}])$
= (by λ-reduction)
$[\![\textbf{the}]\!]([\lambda y_e . y \text{ is dark and } y \text{ is an apple-in-the-row and}$
y is the leftmost element of $\{z : z \text{ is dark and } z \text{ is an apple-in-the-row}\}])$
= (by lexical entry for **the**)
$[\lambda f_{\langle e,t\rangle} : \text{there is exactly one } v \text{ s.t. } f(v) = 1 . \text{ the unique } x \text{ s.t. } f(x) = 1]$
$([\lambda y_e . y \text{ is dark and } y \text{ is an apple-in-the-row and}$
y is the leftmost element of $\{z : z \text{ is dark and } z \text{ is an apple-in-the-row}\}])$
= (by λ-reduction)
the unique x s.t. $[\lambda y_e . y \text{ is dark and } y \text{ is an apple-in-the-row and}$
y is the leftmost element of $\{z : z \text{ is dark and } z \text{ is an apple-in-the-row}\}](x) = 1$,
provided that there is exactly one v s.t. $[\lambda y_e . y \text{ is dark and } y \text{ is an apple-in-the-row}$
and y is the leftmost element of $\{z : z \text{ is dark and } z \text{ is an apple-in-the-row}\}](v) = 1$
= (by λ-reduction)
the unique x s.t. x is dark and x is an apple-in-the-row and
x is the leftmost element of $\{z : z \text{ is dark and } z \text{ is an apple-in-the-row}\}$,
provided that there is exactly one v s.t. v is dark and v is an apple-in-the-row and
v is the leftmost element of $\{z : z \text{ is dark and } z \text{ is an apple-in-the-row}\}$
= (by given scenario)

the unique x s.t. x is dark and x is an apple-in-the-row and x is the leftmost element of $\{b_2, b_4, b_5\}$

= (by given scenario)

b_2

Alex Silk
Heim & Kratzer, Chapter 5 selected exercises

(1) Chapter 5, p. 95

a. $[\![\textbf{Bob [invited t] }]\!]^{m}$ = (by FA)
$[\![\textbf{invited t}]\!]^{m}([\![\textbf{Bob}]\!]^{m})$ = (by FA)
$[\![\textbf{invited}]\!]^{m}([\![\textbf{t}]\!]^{m})([\![\textbf{Bob}]\!]^{m})$ = (by convention (9))
$[\![\textbf{invited}]\!]([\![\textbf{t}]\!]^{m})([\![\textbf{Bob}]\!])$ = (by lexical entries for **invited**, **Bob**)
$[\lambda x_e . [\lambda y_e . y \text{ invited } x]]([\![\textbf{t}]\!]^{m})(\text{Bob})$ = (by (8), p. 92)
$[\lambda x_e . [\lambda y_e . y \text{ invited } x]](m)(\text{Bob})$ = (by λ-reduction)
$[\lambda y_e . y \text{ invited m}](\text{Bob})$ = (by λ-reduction)
1 iff Bob invited m

b. b, j, and m

c. $[\![\textbf{(i)}]\!]$ is undefined because it's not the case that for all assignments a and b, $[\![\textbf{(i)}]\!]^{a} = [\![\textbf{(i)}]\!]^{b}$ (per Definition (9) on p. 94). For example, $[\![\textbf{(i)}]\!]^{m} = 1$, but $[\![\textbf{(i)}]\!]^{s} = 0$ (since it's not the case that b invites s).

(2) Chapter 5, p. 112

$$g = \begin{bmatrix} 1 & \rightarrow & \text{“Barriers”} \\ 2 & \rightarrow & \text{Joe} \end{bmatrix}$$

$[\![\textbf{[he}_2 \textbf{ [wrote [t}_1\textbf{]]]]}]\!]^{g}$ = (by FA)
$[\![\textbf{[wrote [t}_1\textbf{]]}]\!]^{g}([\![\textbf{he}_2]\!]^{g})$ = (by FA)
$[\![\textbf{wrote}]\!]^{g}([\![\textbf{t}_1]\!]^{g})([\![\textbf{he}_2]\!]^{g})$ = (by convention (9) and lexical entry for **write**)
$[\lambda x_e . [\lambda y_e . y \text{ wrote } x]]([\![\textbf{t}_1]\!]^{g})([\![\textbf{he}_2]\!]^{g})$ = (by PT)
$[\lambda x_e . [\lambda y_e . y \text{ wrote } x]](g(1))([\![\textbf{he}_2]\!]^{g})$ = (by PT)
$[\lambda x_e . [\lambda y_e . y \text{ wrote } x]](g(1))(g(2))$ = (by g)
$[\lambda x_e . [\lambda y_e . y \text{ wrote } x]](\text{“Barriers”})(\text{Joe})$ = (by λ-reduction)
$[\lambda y_e . y \text{ wrote “Barriers”}](\text{Joe})$ = (by λ-reduction)
1 iff Joe wrote “Barriers”

(3) Chapter 5, p. 113

$$\begin{bmatrix} 2 & \rightarrow & \text{Joe} \\ 5 & \rightarrow & \text{Mary} \\ 7 & \rightarrow & \text{Ann} \end{bmatrix}^{\text{Mary}/2} = \begin{bmatrix} 2 & \rightarrow & \text{Mary} \\ 5 & \rightarrow & \text{Mary} \\ 7 & \rightarrow & \text{Ann} \end{bmatrix}$$

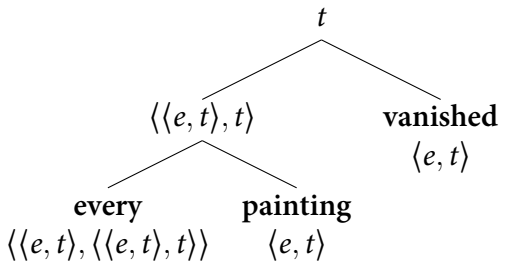
(4) Chapter 5, p. 115

$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and $[\![\textbf{wh}_1 \textbf{ he}_2 \textbf{ wrote t}_1]\!]^{[2 \rightarrow x]}(y) = 1]$
= (by PA)
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and $[\lambda z_e . [\![\textbf{he}_2 \textbf{ wrote t}_1]\!]^{[2 \rightarrow x]^{z/1}}](y) = 1]$
= (by Defn. 13 (assignment modification))
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and $[\lambda z_e . [\![\textbf{he}_2 \textbf{ wrote t}_1]\!]^{\begin{bmatrix} 1 \rightarrow z \\ 2 \rightarrow x \end{bmatrix}}](y) = 1]$
= (by FA)

$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and $[\lambda z_e$. $[\![\textbf{wrote } \textbf{t}_1]\!]^{\left[\begin{smallmatrix}1\to z\\2\to x\end{smallmatrix}\right]}([\![\textbf{he}_2]\!]^{\left[\begin{smallmatrix}1\to z\\2\to x\end{smallmatrix}\right]})](y) = 1]$
= (by FA)
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book
and $[\lambda z_e$. $[\![\textbf{wrote}]\!]^{\left[\begin{smallmatrix}1\to z\\2\to x\end{smallmatrix}\right]}([\![\textbf{t}_1]\!]^{\left[\begin{smallmatrix}1\to z\\2\to x\end{smallmatrix}\right]})([\![\textbf{he}_2]\!]^{\left[\begin{smallmatrix}1\to z\\2\to x\end{smallmatrix}\right]})](y) = 1]$
= (by convention (9) and lexical entry for **write**)
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book
and $[\lambda z_e$. $[\lambda u_e$. $[\lambda v_e$. v wrote $u]]([\![\textbf{t}_1]\!]^{\left[\begin{smallmatrix}1\to z\\2\to x\end{smallmatrix}\right]})([\![\textbf{he}_2]\!]^{\left[\begin{smallmatrix}1\to z\\2\to x\end{smallmatrix}\right]})](y) = 1]$
= (by PT x2)
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and $[\lambda z_e$. $[\lambda u_e$. $[\lambda v_e$. v wrote $u]](z)(x)](y) = 1]$
= (by λ-reduction)
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and $[\lambda z_e$. $[\lambda v_e$. v wrote $z](x)](y) = 1]$
= (by λ-reduction)
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and $[\lambda z_e$. x wrote $z](y) = 1]$
= (by λ-reduction)
$[\lambda x_e$. Mary reviewed the unique y s.t. y is a book and x wrote $y]$

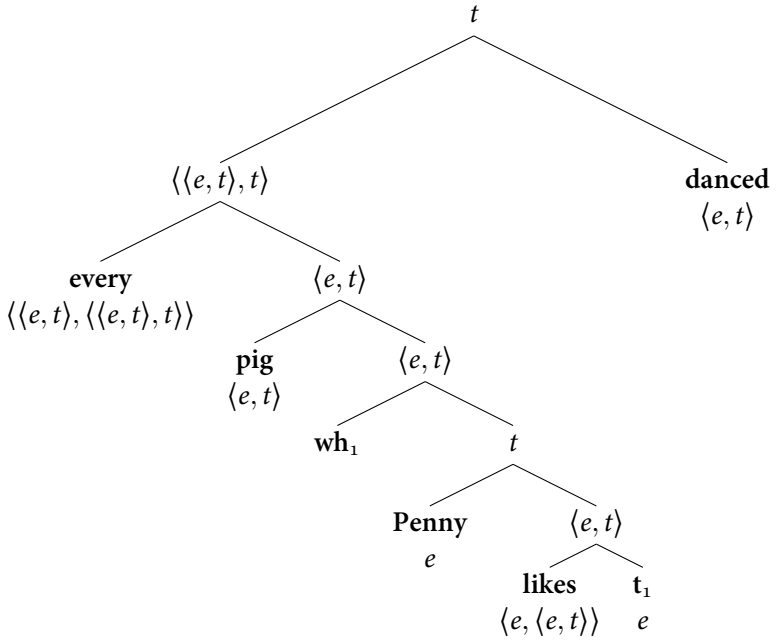
Alex Silk
Heim & Kratzer, Chapters 5, 6 exercises

(1) 'Every painting vanished'



⟦**every painting vanished**⟧
= (by FA)
⟦**every painting**⟧ (⟦**vanished**⟧)
= (by FA)
⟦**every**⟧ (⟦**painting**⟧) (⟦**vanished**⟧)
= (by lexical entry for 'every')
$[\lambda f_{\langle e,t\rangle} . [\lambda g_{\langle e,t\rangle} .$ for all x_e s.t. $f(x) = 1, g(x) = 1]]$(⟦**painting**⟧) (⟦**vanished**⟧)
= (by λ-reduction)
$[\lambda g_{\langle e,t\rangle} .$ for all x_e s.t. ⟦**painting**⟧$(x) = 1, g(x) = 1]$(⟦**vanished**⟧)
= (by λ-reduction)
1 iff for all x_e s.t. ⟦**painting**⟧$(x) = 1$, ⟦**vanished**⟧$(x) = 1$
iff (by lexical entry for 'painting')
for all x_e s.t. $[\lambda y_e . y$ is a painting$](x) = 1$, ⟦**vanished**⟧$(x) = 1$
iff (by lexical entry for 'vanished')
for all x_e s.t. $[\lambda y_e . y$ is a painting$](x) = 1, [\lambda z_e . z$ vanished$](x) = 1$
iff (by λ-reduction)
for all x_e s.t. x is a painting, $[\lambda z_e . z$ vanished$](x) = 1$
iff (by λ-reduction)
for all x_e s.t. x is a painting, x vanished

(2) 'Every pig who Penny likes danced.'

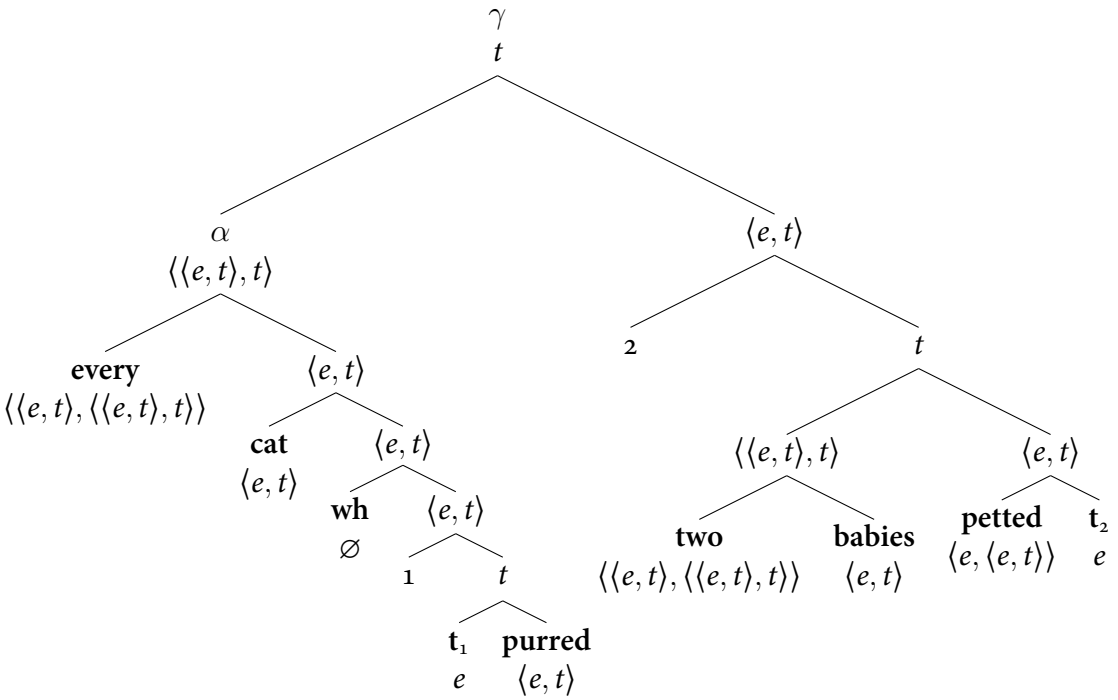


$[\![\textbf{every pig wh}_1\ \textbf{Penny likes t}_1\ \textbf{danced}]\!]$
= (by FA)
$[\![\textbf{every pig wh}_1\ \textbf{Penny likes t}_1]\!]([\![\textbf{danced}]\!])$
= (by FA)
$[\![\textbf{every}]\!]([\![\textbf{pig wh}_1\ \textbf{Penny likes t}_1]\!])([\![\textbf{danced}]\!])$
= (by lexical entry for 'every')
$[\lambda f_{\langle e,t\rangle}\,.\,[\lambda g_{\langle e,t\rangle}\,.\,\text{for all } x_e \text{ s.t. } f(x) = 1, g(x) = 1]]([\![\textbf{pig wh}_1\ \textbf{Penny likes t}_1]\!])([\![\textbf{danced}]\!])$
= (by λ-reduction)
$[\lambda g_{\langle e,t\rangle}\,.\,\text{for all } x_e \text{ s.t. } [\![\textbf{pig wh}_1\ \textbf{Penny likes t}_1]\!](x) = 1, g(x) = 1]([\![\textbf{danced}]\!])$
= (by λ-reduction)
1 iff for all x_e s.t. $[\![\textbf{pig wh}_1\ \textbf{Penny likes t}_1]\!](x) = 1$, $[\![\textbf{danced}]\!](x) = 1$
iff (by lexical entry for 'danced')
for all x_e s.t. $[\![\textbf{pig wh}_1\ \textbf{Penny likes t}_1]\!](x) = 1$, $[\lambda y_e\,.\,y \text{ danced}](x) = 1$
iff (by λ-reduction)
for all x_e s.t. $[\![\textbf{pig wh}_1\ \textbf{Penny likes t}_1]\!](x) = 1$, x danced
iff (by PM)
for all x_e s.t. $[\lambda y_e\,.\,[\![\textbf{pig}]\!](y) = 1 \text{ and } [\![\textbf{wh}_1\ \textbf{Penny likes t}_1]\!](y) = 1](x) = 1$, x danced
iff (by lexical entry for 'pig')
for all x_e s.t. $[\lambda y_e\,.\,[\lambda z_e\,.\,z \text{ is a pig}](y) = 1 \text{ and } [\![\textbf{wh}_1\ \textbf{Penny likes t}_1]\!](y) = 1](x) = 1$, x danced
iff (by λ-reduction)
for all x_e s.t. $[\lambda y_e\,.\,y \text{ is a pig and } [\![\textbf{wh}_1\ \textbf{Penny likes t}_1]\!](y) = 1](x) = 1$, x danced
iff (by λ-reduction)
for all x_e s.t. x is a pig and $[\![\textbf{wh}_1\ \textbf{Penny likes t}_1]\!](x) = 1$, x danced
iff (by convention (11))
for all x_e s.t. x is a pig and $[\![\textbf{wh}_1\ \textbf{Penny likes t}_1]\!]^{\varnothing}(x) = 1$, x danced
iff (by PA)

for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\![\textbf{Penny likes t}_1]\!]^{\varnothing^{y/1}}](x) = 1$, x danced
iff (by assignment modification)
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\![\textbf{Penny likes t}_1]\!]^{[1\to y]}](x) = 1$, x danced
iff (by FA)
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\![\textbf{likes t}_1]\!]^{[1\to y]}([\![\textbf{Penny}]\!]^{[1\to y]})](x) = 1$, x danced
iff (by convention (9) and lexical entry for 'Penny')
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\![\textbf{likes t}_1]\!]^{[1\to y]}(\text{Penny})](x) = 1$, x danced
iff (by FA)
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\![\textbf{likes}]\!]^{[1\to y]}([\![\textbf{t}_1]\!]^{[1\to y]})(\text{Penny})](x) = 1$, x danced
iff (by PT)
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\![\textbf{likes}]\!]^{[1\to y]}(y)(\text{Penny})](x) = 1$, x danced
iff (by convention (9) and lexical entry for 'likes')
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\lambda z_e\,.\,[\lambda b_e\,.\,b \text{ likes } z]](y)(\text{Penny})](x) = 1$, x danced
iff (by λ-reduction)
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,[\lambda b_e\,.\,b \text{ likes } y](\text{Penny})](x) = 1$, x danced
iff (by λ-reduction)
for all x_e s.t. x is a pig and $[\lambda y_e\,.\,\text{Penny likes } y](x) = 1$, x danced
iff (by λ-reduction)
for all x_e s.t. x is a pig and Penny likes x, x danced

Alex Silk
Heim & Kratzer, Chapter 7 samples

(1) 'Two babies petted every cat which purred.'
- intended: "for every x such that x is a cat which purred, there are two babies which petted x"
- (ignore singular/plural differences re 'baby'/'babies')



Sample derivation using a *subproof*:

Subproof A:
$[\![\alpha]\!]$
= (by FA)
$[\![\textbf{every}]\!]([\![\textbf{cat wh 1 t}_1\ \textbf{purred}]\!])$
= (by PM)
$[\![\textbf{every}]\!]([\lambda x_e\,.\,[\![\textbf{cat}]\!](x) = 1 \text{ and } [\![\textbf{wh 1 t}_1\ \textbf{purred}]\!](x) = 1])$
= (by lexical entry for 'cat')
$[\![\textbf{every}]\!]([\lambda x_e\,.\,[\lambda y_e\,.\,y \text{ is a cat}](x) = 1 \text{ and } [\![\textbf{wh 1 t}_1\ \textbf{purred}]\!](x) = 1])$
= (by λ-reduction)
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\![\textbf{wh 1 t}_1\ \textbf{purred}]\!](x) = 1])$
= (by vacuity of 'wh')
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\![\textbf{1 t}_1\ \textbf{purred}]\!](x) = 1])$
= (by convention (11) and PA)
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\lambda y_e\,.\,[\![\textbf{t}_1\ \textbf{purred}]\!]^{\varnothing^{y/1}}](x) = 1])$
= (by assignment modification)
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\lambda y_e\,.\,[\![\textbf{t}_1\ \textbf{purred}]\!]^{[1\to y]}](x) = 1])$
= (by FA)

$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\lambda y_e\,.\,[\![\textbf{purred}]\!]^{[1\to y]}([\![\mathbf{t}_1]\!]^{[1\to y]})](x) = 1])$
= (by PT)
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\lambda y_e\,.\,[\![\textbf{purred}]\!]^{[1\to y]}(y)](x) = 1])$
= (by convention (9) and lexical entry for 'purred')
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\lambda y_e\,.\,[\lambda z_e\,.\,z \text{ purred}](y)](x) = 1])$
= (by λ-reduction)
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } [\lambda y_e\,.\,y \text{ purred}](x) = 1])$
= (by λ-reduction)
$[\![\textbf{every}]\!]([\lambda x_e\,.\,x \text{ is a cat and } x \text{ purred}])$
= (by lexical entry for 'every')
$[\lambda f_{\langle e,t\rangle}\,.\,[\lambda g_{\langle e,t\rangle}\,.\,\text{for all } y \text{ such that } f(y) = 1, g(y) = 1]]([\lambda x_e\,.\,x \text{ is a cat and } x \text{ purred}])$
= (by λ-reduction)
$[\lambda g_{\langle e,t\rangle}\,.\,\text{for all } y \text{ such that } [\lambda x_e\,.\,x \text{ is a cat and } x \text{ purred}](y) = 1, g(y) = 1]$
= (by λ-reduction)
$[\lambda g_{\langle e,t\rangle}\,.\,\text{for all } y \text{ such that } y \text{ is a cat and } y \text{ purred}, g(y) = 1]$

$[\![\gamma]\!]$
= (by FA)
$[\![\alpha]\!]([\![\textbf{2 two babies petted } \mathbf{t}_2]\!])$
= (by convention (11) and PA)
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two babies petted } \mathbf{t}_2]\!]^{\varnothing^{x/2}}])$
= (by assignment modification)
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two babies petted } \mathbf{t}_2]\!]^{[2\to x]}])$
= (by FA)
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two babies}]\!]^{[2\to x]}([\![\textbf{petted } \mathbf{t}_2]\!]^{[2\to x]})])$
= (by FA)
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two babies}]\!]^{[2\to x]}([\![\textbf{petted}]\!]^{[2\to x]}([\![\mathbf{t}_2]\!]^{[2\to x]}))])$
= (by PT)
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two babies}]\!]^{[2\to x]}([\![\textbf{petted}]\!]^{[2\to x]}(x))])$
= (by convention (9) and lexical entry for 'petted')
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two babies}]\!]^{[2\to x]}([\lambda u_e\,.[\lambda v_e\,.\,v \text{ petted } u]](x))])$
= (by λ-reduction)
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two babies}]\!]^{[2\to x]}([\lambda v_e\,.\,v \text{ petted } x])])$
= (by FA)
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two}]\!]^{[2\to x]}([\![\textbf{babies}]\!]^{[2\to x]})([\lambda v_e\,.\,v \text{ petted } x])])$
= (by convention (9) and lexical entry for 'baby')
$[\![\alpha]\!]([\lambda x_e\,.\,[\![\textbf{two}]\!]^{[2\to x]}([\lambda z_e\,.\,z \text{ is a baby}])([\lambda v_e\,.\,v \text{ petted } x])])$
= (by convention (9) and lexical entry for 'two')
$[\![\alpha]\!]([\lambda x_e\,.\,[\lambda f_{\langle e,t\rangle}\,.\,[\lambda g_{\langle e,t\rangle}\,.\,\text{there are at least two } m \text{ such that } f(m) = 1 \text{ and } g(m) = 1]]$
$([\lambda z_e\,.\,z \text{ is a baby}])([\lambda v_e\,.\,v \text{ petted } x])])$
= (by λ-reduction)
$[\![\alpha]\!]([\lambda x_e\,.\,[\lambda g_{\langle e,t\rangle}\,.\,\text{there are at least two } m \text{ such that } [\lambda z_e\,.\,z \text{ is a baby}](m) = 1 \text{ and } g(m) = 1]$
$([\lambda v_e\,.\,v \text{ petted } x])])$
= (by λ-reduction)

$[\![\alpha]\!]([\lambda x_e\,.\,[\lambda g_{\langle e,t\rangle}\,.$ there are at least two m such that m is a baby and $g(m) = 1]$
$([\lambda v_e\,.\,v$ petted $x])])$
= (by λ-reduction)
$[\![\alpha]\!]([\lambda x_e\,.$ there are at least two m such that m is a baby and $[\lambda v_e\,.\,v$ petted $x](m) = 1])$
= (by λ-reduction)
$[\![\alpha]\!]([\lambda x_e\,.$ there are at least two m such that m is a baby and m petted $x])$
= (by Subproof A)
$[\lambda g_{\langle e,t\rangle}\,.$ for all y such that y is a cat and y purred, $g(y) = 1]([\lambda x_e\,.$ there are at least two m such that m is a baby and m petted $x])$
= (by λ-reduction)
1 iff for all y such that y is a cat and y purred, $[\lambda x_e\,.$ there are at least two m such that m is a baby and m petted $x](y) = 1$
iff (by λ-reduction)
for all y such that y is a cat and y purred, there are at least two m such that m is a baby and m petted y