1 Existential quantification

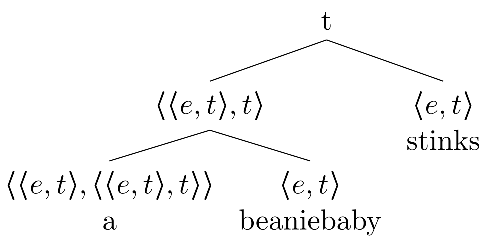
⟦a beaniebaby stinks⟧

Lexical entries:

1. ⟦beaniebaby⟧ = $\lambda y[BEANIEBABY(y)]$
2. ⟦stink⟧ = $\lambda z[STINK(z)]$
3. ⟦a⟧ = $\lambda f_{\langle e,t\rangle}[\lambda g_{\langle e,t\rangle}[\exists x[f(x) \wedge g(x)]]]$

★ If you used other variables for any lexical entry, that's also right! e.g., $\lambda z[BEANIEBABY(z)] = \lambda y[BEANIEBABY(y)]$, etc.

Tree annotated with types:



★ Common mistake with types: be careful with how many brackets you have, and where!

- ➥ $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle$: ill-formed! A closing bracket is missing.
- ➥ $\langle e,t,t\rangle$: ill-formed! There should be one input, one output for each pair of brackets (e.g., $\langle\langle e,t\rangle,t\rangle$).
- ➥ $\langle e,\langle t,t\rangle\rangle$: not the same thing as $\langle\langle e,t\rangle,t\rangle$!
- ➥ $\langle\langle\langle e,t\rangle,\langle e,t\rangle\rangle,t\rangle$: not the same thing as $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$!

Step-by-step computation:

1. ⟦a beaniebaby⟧
   (a) $= [\![\text{a}]\!]([\![\text{beaniebaby}]\!])$
   (b) $= \lambda f_{\langle e,t\rangle}[\lambda g_{\langle e,t\rangle}[\exists x[f(x) \wedge g(x)]]]([\![\text{beaniebaby}]\!])$
   (c) $= \lambda g_{\langle e,t\rangle}[\exists x[[\![\text{beaniebaby}]\!](x) \wedge g(x)]]$
   (d) $= \lambda g_{\langle e,t\rangle}[\exists x[\lambda y[BEANIEBABY(y)](x) \wedge g(x)]]$
   (e) $= \lambda g_{\langle e,t\rangle}[\exists x[BEANIEBABY(x) \wedge g(x)]]$
2. ⟦a beaniebaby stinks⟧
   (a) $= [\![\text{a beaniebaby stink}]\!]$
   (b) $= [\![\text{a beaniebaby}]\!]([\![\text{stink}]\!])$
   (c) $= \lambda g_{\langle e,t\rangle}[\exists x[BEANIEBABY(x) \wedge g(x)]]\ ([\![\text{stink}]\!])$
   (d) $= \exists x[BEANIEBABY(x) \wedge [\![\text{stink}]\!](x)]$
   (e) $= \exists x[BEANIEBABY(x) \wedge \lambda z[STINK(z)](x)]$
   (f) $=$ T iff $\exists x[BEANIEBABY(x) \wedge STINK(x)]$

   ★ Common mistake: Don't forget to state the very final step as a truth condition! Remember, what you're calculating is the denotation of a sentence, which is its truth value/condition.

2 Types summary

1. $\langle e, t\rangle$: type for one-place predicates
2. $\langle\langle e, t\rangle, t\rangle$: type for quantifiers (also sometimes called generalized quantifiers[1]); e.g., *every beaniebaby*, *everyone*, *a/some beaniebaby*, *someone*, *all beaniebabies*, *few/most beaniebabies*, etc.)
3. $\langle\langle e, t\rangle, \langle\langle e, t\rangle, t\rangle\rangle$: type for quantificational determiners; e.g., *every*, *all*, *a*, *some*, *few*, *most*, etc.

What other types are possible? What might that kind of function do?

[1] It's called "generalized" because at a certain point in history, logicians/semanticists discovered that all quantificational phrases (e.g., many beaniebabies, two beaniebabies, few beaniebabies) are all the same thing: a relation between sets. The treatment of those quantifiers was "generalized" from the analysis of ∀ and ∃, hence, "generalized".