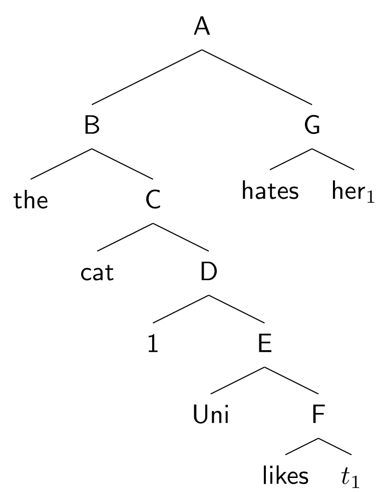
Homework for Friday October 10, 2014

1 More practice with λs

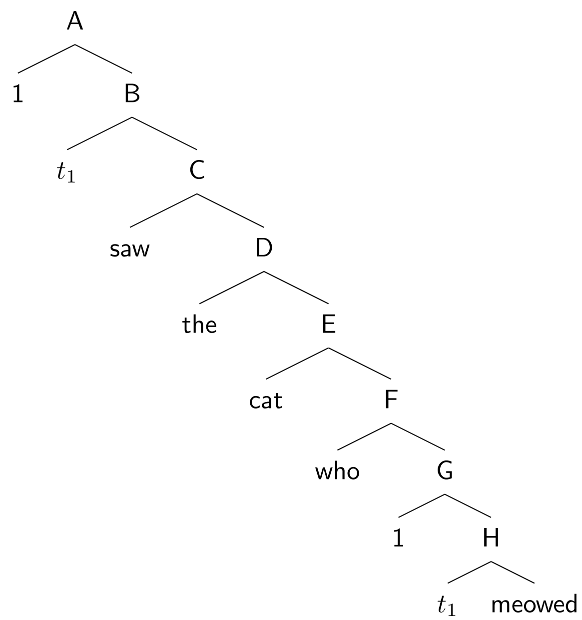
$$\begin{aligned}
(\lambda m.\lambda n.m(\lambda f.n(\lambda x.f(x))))(\lambda k.k(\text{left}'))(\lambda k.k(x)) &= (\lambda n.(\lambda k.k(\text{left}'))(\lambda f.n(\lambda x.f(x))))(\lambda k.k(x)) && \beta\\
&= (\lambda k.k(\text{left}'))(\lambda f.(\lambda k.k(x))(\lambda x.f(x))) && \beta\\
&= (\lambda f.(\lambda k.k(x))(\lambda x.f(x)))(\text{left}') && \beta\\
&= (\lambda k.k(x))(\lambda x.\text{left}'(x)) && \beta\\
&= (\lambda x.\text{left}'(x))(x) && \beta\\
&= \text{left}'(x) && \beta
\end{aligned}$$

2 Relative clauses

- the cat Uni likes hates her:



$$\begin{aligned}
\llbracket \mathsf{A} \rrbracket^g &= \llbracket \mathsf{G} \rrbracket^g(\llbracket \mathsf{B} \rrbracket^g) && \textbf{FA}\\
&= \llbracket \mathsf{G} \rrbracket^g(\llbracket \mathsf{the} \rrbracket^g(\llbracket \mathsf{C} \rrbracket^g)) && \textbf{FA}\\
&= \llbracket \mathsf{G} \rrbracket^g(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\llbracket \mathsf{cat} \rrbracket^g(x) = \llbracket \mathsf{D} \rrbracket^g(x) = 1)) && \textbf{PM}\\
&= \llbracket \mathsf{G} \rrbracket^g(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\llbracket \mathsf{cat} \rrbracket^g(x) = (\lambda y.\llbracket \mathsf{E} \rrbracket^{g[y/1]})(x) = 1)) && \textbf{PA}\\
&= \llbracket \mathsf{G} \rrbracket^g(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\llbracket \mathsf{cat} \rrbracket^g(x) = \llbracket \mathsf{E} \rrbracket^{g[x/1]} = 1)) && \beta\\
&= \llbracket \mathsf{G} \rrbracket^g(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\llbracket \mathsf{cat} \rrbracket^g(x) = \llbracket \mathsf{F} \rrbracket^{g[x/1]}(\llbracket \mathsf{Uni} \rrbracket^{g[x/1]}) = 1)) && \textbf{FA}\\
&= \llbracket \mathsf{G} \rrbracket^g(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\llbracket \mathsf{cat} \rrbracket^g(x) = \llbracket \mathsf{likes} \rrbracket^{g[x/1]}(\llbracket t_1 \rrbracket^{g[x/1]})(\llbracket \mathsf{Uni} \rrbracket^{g[x/1]}) = 1)) && \textbf{FA}\\
&= \llbracket \mathsf{hates} \rrbracket^g(\llbracket \mathsf{her}_1 \rrbracket^g)(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\llbracket \mathsf{cat} \rrbracket^g(x) = \llbracket \mathsf{likes} \rrbracket^{g[x/1]}(\llbracket t_1 \rrbracket^{g[x/1]})(\llbracket \mathsf{Uni} \rrbracket^{g[x/1]}) = 1)) && \textbf{FA}\\
&= \text{hates}'(\llbracket \mathsf{her}_1 \rrbracket^g)(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\,\text{cat}'(x) = \text{likes}'(\llbracket t_1 \rrbracket^{g[x/1]})(\text{u}) = 1)) && \text{Lexicon}\\
&= \text{hates}'(g(1))(\llbracket \mathsf{the} \rrbracket^g(\lambda x.\,\text{cat}'(x) = \text{likes}'(x)(\text{u}) = 1)) && \text{Variables}\\
&= \text{hates}'(g(1))((\lambda P.\iota x.P(x))(\lambda x.\,\text{cat}'(x) = \text{likes}'(x)(\text{u}) = 1)) && \text{Lexicon}\\
&= \text{hates}'(g(1))(\iota x.\,\text{cat}'(x) = \text{likes}'(x)(\text{u}) = 1) && \beta
\end{aligned}$$



$$
\begin{aligned}
\llbracket \mathsf{A} \rrbracket^{g} &= \lambda x. \llbracket \mathsf{B} \rrbracket^{g[x/1]} && \textbf{PA}\\
&= \lambda x. \llbracket \mathsf{C} \rrbracket^{g[x/1]}(\llbracket t_1 \rrbracket^{g[x/1]}) && \textbf{FA}\\
&= \lambda x. \llbracket \mathsf{C} \rrbracket^{g[x/1]}(x) && \text{Variables}\\
&= \lambda x. \llbracket \mathsf{saw} \rrbracket^{g[x/1]}(\llbracket \mathsf{D} \rrbracket^{g[x/1]})(x) && \textbf{FA}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{D} \rrbracket^{g[x/1]})(x) && \text{Lexicon}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\llbracket \mathsf{E} \rrbracket^{g[x/1]}))(x) && \textbf{FA}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y. \llbracket \mathsf{cat} \rrbracket^{g[x/1]}(y) = \llbracket \mathsf{F} \rrbracket^{g[x/1]}(y) = 1))(x) && \textbf{PM}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = \llbracket \mathsf{F} \rrbracket^{g[x/1]}(y) = 1))(x) && \text{Lexicon}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = \llbracket \mathsf{who} \rrbracket^{g[x/1]}(\llbracket \mathsf{G} \rrbracket^{g[x/1]})(y) = 1))(x) && \textbf{FA}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = (\lambda P.P)(\llbracket \mathsf{G} \rrbracket^{g[x/1]})(y) = 1))(x) && \text{Lexicon}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = (\llbracket \mathsf{G} \rrbracket^{g[x/1]})(y) = 1))(x) && \beta\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = (\lambda y. \llbracket \mathsf{H} \rrbracket^{g[x/1][y/1]})(y) = 1))(x) && \textbf{PA}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = \llbracket \mathsf{H} \rrbracket^{g[x/1][y/1]} = 1))(x) && \beta\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = \llbracket \mathsf{meowed} \rrbracket^{g[x/1][y/1]}(\llbracket t_1 \rrbracket^{g[x/1][y/1]}) = 1))(x) && \textbf{FA}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = \mathrm{meowed}'(\llbracket t_1 \rrbracket^{g[x/1][y/1]}) = 1))(x) && \text{Lexicon}\\
&= \lambda x.\, \mathrm{saw}'(\llbracket \mathsf{the} \rrbracket^{g[x/1]}(\lambda y.\, \mathrm{cat}'(y) = \mathrm{meowed}'(y) = 1))(x) && \text{Variables}\\
&= \lambda x.\, \mathrm{saw}'((\lambda P. \iota y. P(y))(\lambda y.\, \mathrm{cat}'(y) = \mathrm{meowed}'(y) = 1))(x) && \text{Lexicon}\\
&= \lambda x.\, \mathrm{saw}'(\iota y. (\lambda y.\, \mathrm{cat}'(y) = \mathrm{meowed}'(y) = 1)(y))(x) && \beta\\
&= \lambda x.\, \mathrm{saw}'(\iota y.\, \mathrm{cat}'(y) = \mathrm{meowed}'(y) = 1)(x) && \beta
\end{aligned}
$$

▷ We can sum up the state of affairs as follows: an abstraction index binds a pronoun/trace iff the number on the abstraction index is the same as the number on the pronoun/trace, the abstraction index c-commands the co-indexed pronoun/trace, and no other c-commanding, co-indexed abstraction index intervenes between the abstraction index and the pronoun/trace. Thus, her_1 is not bound in the first tree (since it is co-indexed but not c-commanded by the abstraction index), and the abstraction indices in the second tree bind the nearest traces.

3 Quantifiers

- Meanings for quantificational DPs:

  1. $[\![\textsf{not every phonologist}]\!]^g = \lambda P.\,\text{phonologist}' \not\subseteq P$
  2. $[\![\textsf{three out of four dentists}]\!]^g = \lambda P.\,\frac{|\text{dentist}' \cap P|}{|\text{dentist}'|} \geq \frac{3}{4}$

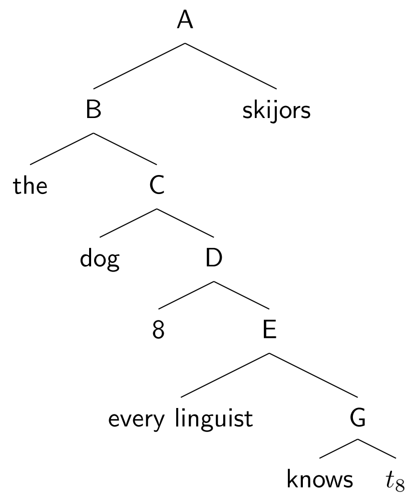
     ▷ Note: we tend to hear sentences like three out of four dentists agree as entailing that *no more than* three out of four agree. This is plausibly an implicature. Suppose I say if three out of four dentists agree, this toothbrush gets approved. Surely if four out of four approve, the toothbrush still gets approved.

  3. $[\![\textsf{every linguist except John}]\!]^g = \lambda P.\,(\text{linguist}' - \{\text{j}\}) \subseteq P$, and $\text{j} \notin P$

     ▷ Note: this construction seems to **presuppose** that John is a linguist. If you like, you can write that into the meaning in terms of a definedness condition. Notice also that the second condition ($\{\text{j}\} \notin P$) is crucial. Without it, we do not end up entailing that John isn't among the people of whom P holds. Do you see why?

  4. $[\![\textsf{at least four but no more than ten hotels}]\!]^g = \lambda P.\,4 \leq |\text{hotel}' \cap P| \leq 10$
  5. $[\![\textsf{more than ten or fewer than five semanticists}]\!]^g = \lambda P.\,|\text{sems}' \cap P| > 10$, or $|\text{sems}' \cap P| < 5$

- the dog every linguist knows skijors:

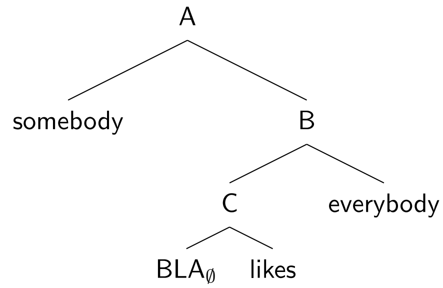


$$\begin{aligned}
[\![\mathsf{A}]\!]^g &= [\![\mathsf{skijors}]\!]^g([\![\mathsf{B}]\!]^g) && \textbf{FA} \\
&= \text{skijors}'([\![\mathsf{B}]\!]^g) && \text{Lexicon} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g([\![\mathsf{C}]\!]^g)) && \textbf{FA} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.[\![\mathsf{dog}]\!]^g(x) = [\![\mathsf{D}]\!]^g(x) = 1)) && \textbf{PM} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = [\![\mathsf{D}]\!]^g(x) = 1)) && \text{Lexicon} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = (\lambda x.[\![\mathsf{E}]\!]^{g[x/8]})(x) = 1)) && \textbf{PA} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = [\![\mathsf{E}]\!]^{g[x/8]} = 1)) && \beta \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = [\![\mathsf{every\ linguist}]\!]^{g[x/8]}([\![\mathsf{G}]\!]^{g[x/8]}) = 1)) && \textbf{FA} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = [\![\mathsf{every\ linguist}]\!]^{g[x/8]}([\![\mathsf{knows}]\!]^{g[x/8]}([\![t_8]\!]^{g[x/8]})) = 1)) && \textbf{FA} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = [\![\mathsf{every\ linguist}]\!]^{g[x/8]}(\text{knows}'([\![t_8]\!]^{g[x/8]})) = 1)) && \text{Lexicon} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = [\![\mathsf{every\ linguist}]\!]^{g[x/8]}(\text{knows}'(x)) = 1)) && \text{Variables} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = (\lambda P.\,\text{ling}' \subseteq \{y : P(y)\})(\text{knows}'(x)) = 1)) && \text{Lexicon} \\
&= \text{skijors}'([\![\mathsf{the}]\!]^g(\lambda x.\,\text{dog}'(x) = \text{ling}' \subseteq \{y : \text{knows}'(x)(y)\} = 1)) && \beta \\
&= \text{skijors}'((\lambda P.\iota x.P(x))(\lambda x.\,\text{dog}'(x) = \text{ling}' \subseteq \{y : \text{knows}'(x)(y)\} = 1)) && \text{Lexicon} \\
&= \text{skijors}'(\iota x.(\lambda x.\,\text{dog}'(x) = \text{ling}' \subseteq \{y : \text{knows}'(x)(y)\} = 1)(x)) && \beta \\
&= \text{skijors}'(\iota x.\,\text{dog}'(x) = \text{ling}' \subseteq \{y : \text{knows}'(x)(y)\} = 1) && \beta
\end{aligned}$$

- Type $\langle\langle e,t\rangle,t\rangle$ meanings for New Jersey and the Queen of England:

  1. $\lambda P.P(\text{nj})$
  2. $\lambda P.P(\text{qoe})$

- Using $\mathsf{BLA}_\emptyset$ to derive a meaning for the ambiguous sentence somebody likes everybody without QR. This is the surface-scope interpretation. $\mathsf{BLA}_\emptyset$ by itself (and without QR) doesn't give us a way to derive the inverse-scope interpretation:



$$
\begin{aligned}
[\![\mathsf{A}]\!]^g &= [\![\mathsf{somebody}]\!]^g([\![\mathsf{B}]\!]^g) && \textbf{FA}\\
&= [\![\mathsf{somebody}]\!]^g([\![\mathsf{C}]\!]^g([\![\mathsf{everybody}]\!]^g)) && \textbf{FA}\\
&= [\![\mathsf{somebody}]\!]^g([\![\mathsf{BLA}_\emptyset]\!]^g([\![\mathsf{likes}]\!]^g)([\![\mathsf{everybody}]\!]^g)) && \textbf{FA}\\
&= [\![\mathsf{somebody}]\!]^g((\lambda R.\lambda \mathcal{Q}.\lambda x.\mathcal{Q}(\lambda y.R(y)(x)))(\text{likes}')([\![\mathsf{everybody}]\!]^g)) && \text{Lexicon}\\
&= [\![\mathsf{somebody}]\!]^g((\lambda \mathcal{Q}.\lambda x.\mathcal{Q}(\lambda y.\text{likes}'(y)(x)))([\![\mathsf{everybody}]\!]^g)) && \beta\\
&= [\![\mathsf{somebody}]\!]^g(\lambda x.[\![\mathsf{everybody}]\!]^g(\lambda y.\text{likes}'(y)(x))) && \beta\\
&= [\![\mathsf{somebody}]\!]^g(\lambda x.(\lambda P.\,\text{ppl}' \subseteq \{y : P(y)\})(\lambda y.\text{likes}'(y)(x))) && \text{Lexicon}\\
&= [\![\mathsf{somebody}]\!]^g(\lambda x.\,\text{ppl}' \subseteq \{y : (\lambda y.\text{likes}'(y)(x))(y)\}) && \beta\\
&= [\![\mathsf{somebody}]\!]^g(\lambda x.\,\text{ppl}' \subseteq \{y : \text{likes}'(y)(x)\}) && \beta\\
&= (\lambda P.\,\text{ppl}' \cap \{x : P(x)\} \neq \emptyset)(\lambda x.\,\text{ppl}' \subseteq \{y : \text{likes}'(y)(x)\}) && \text{Lexicon}\\
&= \text{ppl}' \cap \{x : (\lambda x.\,\text{ppl}' \subseteq \{y : \text{likes}'(y)(x)\})(x)\} \neq \emptyset && \beta\\
&= \text{ppl}' \cap \{x : \text{ppl}' \subseteq \{y : \text{likes}'(y)(x)\}\} \neq \emptyset && \beta
\end{aligned}
$$

- In prose: there are people x such that every person likes x.

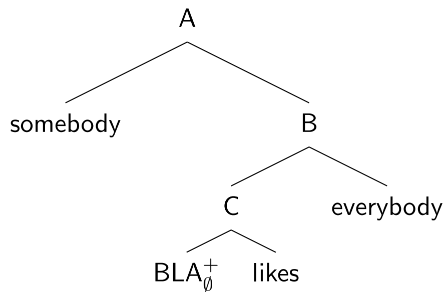
4 Bonus (not required)

- A silent lexical item that allows you to assign the other interpretation to *somebody likes everybody*:

$$[\![\mathsf{BLA}^+_\emptyset]\!]^g := \lambda R_{\langle e,\langle e,t\rangle\rangle}.\lambda \mathcal{Q}_{\langle\langle e,t\rangle,t\rangle}.\lambda \mathcal{P}_{\langle\langle e,t\rangle,t\rangle}.\mathcal{Q}(\lambda y.\mathcal{P}(\lambda x.R(y)(x)))$$

Notice that this is η-equivalent to the following:

$$\lambda R.\lambda \mathcal{Q}.\lambda \mathcal{P}.\mathcal{Q}(\lambda y.\mathcal{P}(R(y)))$$



$$
\begin{aligned}
[\![\mathsf{A}]\!]^g &= [\![\mathsf{B}]\!]^g([\![\mathsf{somebody}]\!]^g) && \textbf{FA}\\
&= [\![\mathsf{C}]\!]^g([\![\mathsf{everybody}]\!]^g)([\![\mathsf{somebody}]\!]^g) && \textbf{FA}\\
&= [\![\mathsf{BLA}^{+}_{\emptyset}]\!]^g([\![\mathsf{likes}]\!]^g)([\![\mathsf{everybody}]\!]^g)([\![\mathsf{somebody}]\!]^g) && \textbf{FA}\\
&= (\lambda R.\lambda \mathcal{Q}.\lambda \mathcal{P}.\mathcal{Q}(\lambda y.\mathcal{P}(\lambda x.R(y)(x))))(\text{likes}')([\![\mathsf{everybody}]\!]^g)([\![\mathsf{somebody}]\!]^g) && \text{Lexicon}\\
&= (\lambda \mathcal{Q}.\lambda \mathcal{P}.\mathcal{Q}(\lambda y.\mathcal{P}(\lambda x.\text{likes}'(y)(x))))([\![\mathsf{everybody}]\!]^g)([\![\mathsf{somebody}]\!]^g) && \beta\\
&= (\lambda \mathcal{P}.[\![\mathsf{everybody}]\!]^g(\lambda y.\mathcal{P}(\lambda x.\text{likes}'(y)(x))))([\![\mathsf{somebody}]\!]^g) && \beta\\
&= [\![\mathsf{everybody}]\!]^g(\lambda y.[\![\mathsf{somebody}]\!]^g(\lambda x.\text{likes}'(y)(x))) && \beta\\
&= [\![\mathsf{everybody}]\!]^g(\lambda y.(\lambda P.\,\text{ppl}' \cap \{x : P(x)\} \neq \emptyset)(\lambda x.\text{likes}'(y)(x))) && \text{Lexicon}\\
&= [\![\mathsf{everybody}]\!]^g(\lambda y.\,\text{ppl}' \cap \{x : (\lambda x.\text{likes}'(y)(x))(x)\} \neq \emptyset) && \beta\\
&= [\![\mathsf{everybody}]\!]^g(\lambda y.\,\text{ppl}' \cap \{x : \text{likes}'(y)(x)\} \neq \emptyset) && \beta\\
&= (\lambda P.\,\text{ppl}' \subseteq \{y : P(y)\})(\lambda y.\,\text{ppl}' \cap \{x : \text{likes}'(y)(x)\} \neq \emptyset) && \text{Lexicon}\\
&= \text{ppl}' \subseteq \{y : (\lambda y.\,\text{ppl}' \cap \{x : \text{likes}'(y)(x)\} \neq \emptyset)(y)\} && \beta\\
&= \text{ppl}' \subseteq \{y : \text{ppl}' \cap \{x : \text{likes}'(y)(x)\} \neq \emptyset\} && \beta
\end{aligned}
$$

- In prose: for every person y, there are people who like y.