Lambda Calculus

$(\lambda x.M)$: define a function for x that returns M.

$f(x) = M \Rightarrow (\lambda x.M)(x) = M$.

β reduction: $((\lambda x.M)N) \rightarrow (M[x := N])$

(in contrast to α-conversation)

e.g. $\lambda x.f(x)$

$\lambda F.\forall y.F(y)$

$(\lambda F.\forall y.F(y))(\lambda x.f(x)) = ?$

$(\lambda x.\ M)\ (N)$

$(\lambda F.\forall y.F(y))(\lambda x.f(x))$

$\Longleftrightarrow_\beta$

$\forall y.(\lambda x.f(x))(y)$.

$\Longleftrightarrow_\beta$ $\forall y.f(y)$.

Q2: Macros: @lambda(X, F) $\lambda X.F$

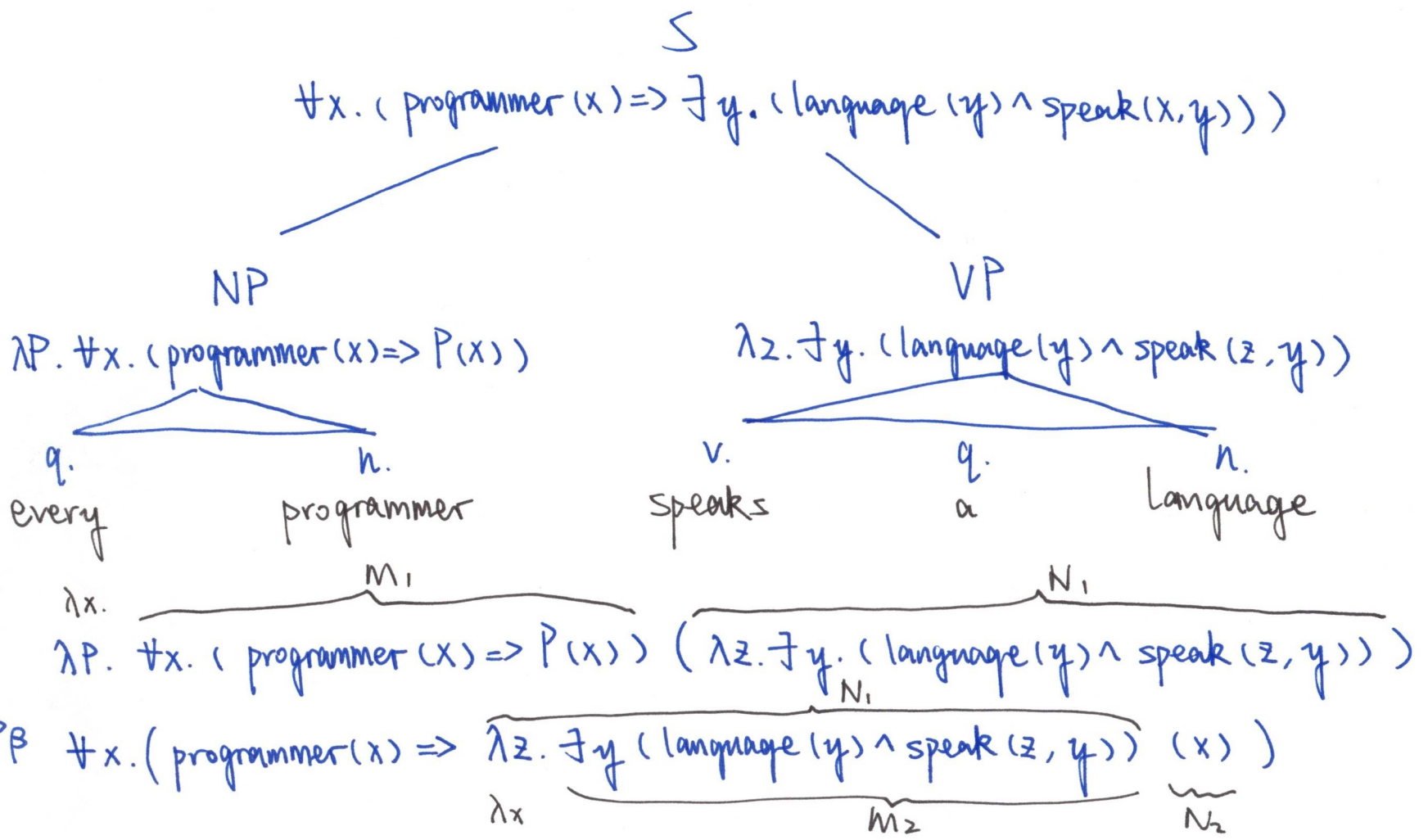
@forall(X, F, G) $\forall X.F \Rightarrow G$

@exists(X, F, G) $\exists X.F \wedge G$

@apply(F, [X, Y, Z]) $F(X, Y, Z)$.

beta-normalize: beta_normalize(@apply(F, [G]), B).

Example: Every programmer speaks a language.



Quantifier Storage

Storage: $\lambda G.\exists y.(\text{language}(y) \wedge G(y))$

$\Downarrow$

$\lambda F.F(z)\ ,\ \text{QSTORE}\langle z; \lambda G.\exists y.(\text{language}(y) \wedge G(y))\rangle$

Retrieval: $\#_{\forall x.}\ \text{programmer}(x) \Rightarrow \text{speaks}(x,z)\ ,\ \text{QSTORE}\langle z; \lambda G.\exists y.(\text{language}(y) \wedge G(y))\rangle$

$\Downarrow$

$\lambda z.\forall x.\ \text{programmer}(x) \Rightarrow \text{speaks}(x,z)\ ,\ \text{QSTORE}\langle \ldots.\rangle$

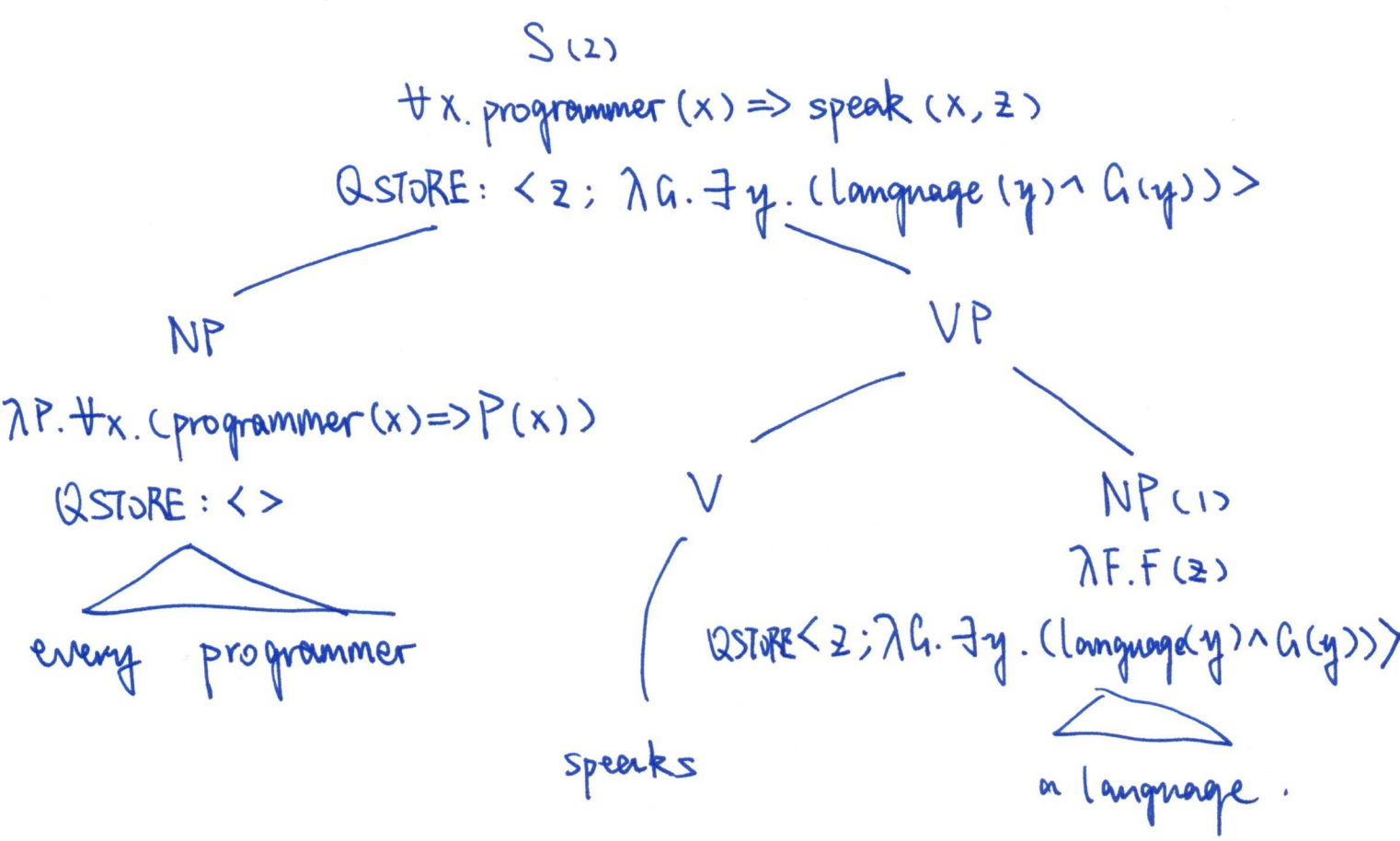
$\Downarrow$

$$\underbrace{\lambda x}\ \overbrace{(\lambda G.\exists y.(\text{language}(y) \wedge G(y)))}^{M_1}\ \overbrace{(\lambda z.\forall x.\ \text{programmer}(x) \Rightarrow \text{speaks}(x,z))}^{N_1}$$

$$\Leftrightarrow_\beta\ \exists y.(\text{language}(y) \wedge \overbrace{\lambda z.\forall x.\ \text{programmer}(x) \Rightarrow \text{speaks}(x,z)}^{M_2}\ \overbrace{(y)}^{N_2})$$

$$\Leftrightarrow_\beta\ \exists y.(\text{language}(y) \wedge \forall x.\ \text{programmer}(x) \Rightarrow \text{speaks}(x,y))$$

Quantifier Storage



STORE at (1), Retrieve at (2)

qaction(Logic, QStore, NewLogic, NewQStore):

↳ start with Logic and QSTORE,
comes out with NewLogic and NewQStore

retrieve(QStore, Logic, NewQstore, NewLogic)

↳ start with QStore and Logic

comes out as NewQStore, what's out is added to Logic
→ NewLogic