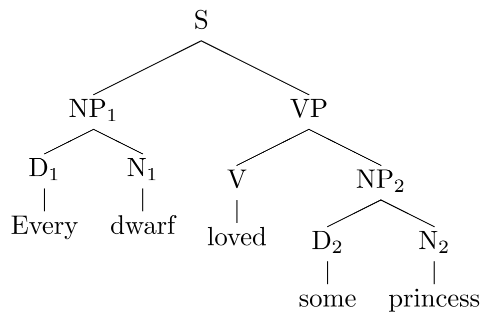
We want to compute the meaning of the sentence "Every dwarf loved some princess". This sentence has the following syntactic structure:



We have the following basic (non-continuized) meanings:

$$
\begin{aligned}
[\![\mathrm{D}_1]\!] = [\![\text{every}]\!] &= \lambda P \lambda Q. \forall x. P(x) \rightarrow Q(x) \\
[\![\mathrm{N}_1]\!] = [\![\text{dwarf}]\!] &= \lambda x.\mathsf{Dwarf}(x) \text{ (or just } \mathsf{Dwarf}) \\
[\![\mathrm{V}]\!] \;\; = [\![\text{loved}]\!] &= \lambda y \lambda x.\mathsf{Love}(x, y) \text{ (or just } \mathsf{Love}) \\
[\![\mathrm{D}_2]\!] = [\![\text{some}]\!] &= \lambda P \lambda Q. \exists y. P(y) \wedge Q(y) \\
[\![\mathrm{N}_2]\!] = [\![\text{princess}]\!] &= \lambda x.\mathsf{Princess}(x) \text{ (or just } \mathsf{Princess})
\end{aligned}
$$

In addition, we define the following functions:

$\mathrm{pure}(u) = \lambda k.k(u)$[1]
$u \texttt{<*>} v = \lambda k.v(\lambda n.u(\lambda m.k(m(n))))$[2]
$v \texttt{>>=} u = \lambda c.v(\lambda a.(u(a))(c))$

We then have the following continuized grammar:

S → NP VP	$\overline{\mathrm{S}} = \overline{\mathrm{VP}}\texttt{<*>}\overline{\mathrm{NP}}$
VP → V NP	$\overline{\mathrm{VP}} = \overline{\mathrm{V}}\texttt{<*>}\overline{\mathrm{NP}}$
NP → D N	$\overline{\mathrm{NP}} = \overline{\mathrm{N}}\texttt{>>=}[\![\mathrm{D}]\!]$
V → loved	$\overline{\mathrm{V}} = \mathrm{pure}([\![\text{loved}]\!])$
N → dwarf	$\overline{\mathrm{N}} = \mathrm{pure}([\![\text{dwarf}]\!])$
N → princess	$\overline{\mathrm{N}} = \mathrm{pure}([\![\text{princess}]\!])$
D → every	$[\![\mathrm{D}]\!] = [\![\text{every}]\!]$
D → some	$[\![\mathrm{D}]\!] = [\![\text{some}]\!]$

[1] van Eijck and Unger call this `cpsConst`; one could also call this `return`
[2] van Eijck and Unger call this `cpsApply`

First, we have $\overline{\text{N}_2} = \text{pure}([\![\text{princess}]\!]) = \lambda k.k(\mathsf{Princess})$. Then we can compute $\overline{\text{NP}_2} = \overline{\text{N}_2}\texttt{>>=}[\![\text{D}_2]\!]$ as follows:

$$\begin{aligned}
\overline{\text{NP}_2} = \overline{\text{some princess}} &= \overline{\text{N}_2}\texttt{>>=}[\![\text{D}_2]\!] \\
&= \lambda c.\overline{\text{N}_2}(\lambda a.([\![\text{D}_2]\!](a))(c)) \\
&= \lambda c.(\lambda k.k(\mathsf{Princess}))(\lambda a.([\![\text{D}_2]\!](a))(c)) \\
&= \lambda c.(\lambda a.([\![\text{D}_2]\!](a))(c))(\mathsf{Princess}) \\
&= \lambda c.([\![\text{D}_2]\!](\mathsf{Princess}))(c) \\
&= \lambda c.((\lambda P\lambda Q.\exists y.P(y) \wedge Q(y))(\mathsf{Princess}))(c) \\
&= \lambda c.(\lambda Q.\exists y.\mathsf{Princess}(y) \wedge Q(y))(c) \\
&= \lambda c.\exists y.\mathsf{Princess}(y) \wedge c(y)
\end{aligned}$$

Then, we have $\overline{\text{V}} = \text{pure}([\![\text{loved}]\!]) = \lambda n.n(\mathsf{Love})$. Then we can compute $\overline{\text{VP}} = \overline{\text{V}}\texttt{<*>}\overline{\text{NP}_2}$ as follows:

$$\begin{aligned}
\overline{\text{VP}} = \overline{\text{loved some princess}} &= \overline{\text{V}}\texttt{<*>}\overline{\text{NP}_2} \\
&= \lambda k.\overline{\text{NP}_2}(\lambda n.\overline{\text{V}}(\lambda m.k(m(n)))) \\
&= \lambda k.(\lambda c.\exists y.\mathsf{Princess}(y) \wedge c(y))(\lambda n.\overline{\text{V}}(\lambda m.k(m(n)))) \\
&= \lambda k.\exists y.\mathsf{Princess}(y) \wedge (\lambda n.\overline{\text{V}}(\lambda m.k(m(n))))(y) \\
&= \lambda k.\exists y.\mathsf{Princess}(y) \wedge \overline{\text{V}}(\lambda m.k(m(y))) \\
&= \lambda k.\exists y.\mathsf{Princess}(y) \wedge (\lambda n.n(\mathsf{Love}))(\lambda m.k(m(y))) \\
&= \lambda k.\exists y.\mathsf{Princess}(y) \wedge (\lambda m.k(m(y)))(\mathsf{Love}) \\
&= \lambda k.\exists y.\mathsf{Princess}(y) \wedge k(\mathsf{Love}(y))
\end{aligned}$$

Similarly to $\overline{\text{N}_2}$, we have $\overline{\text{N}_1} = \text{pure}([\![\text{dwarf}]\!]) = \lambda k.k(\mathsf{Dwarf})$. In the same way, we can compute $\overline{\text{NP}_1} = \overline{\text{N}_1}\texttt{>>=}[\![\text{D}_1]\!]$ as follows:

$$\begin{aligned}
\overline{\text{NP}_1} = \overline{\text{every dwarf}} &= \overline{\text{N}_1}\texttt{>>=}[\![\text{D}_1]\!] \\
&= \lambda c.\overline{\text{N}_1}(\lambda a.([\![\text{D}_1]\!](a))(c)) \\
&= \lambda c.(\lambda k.k(\mathsf{Dwarf}))(\lambda a.([\![\text{D}_1]\!](a))(c)) \\
&= \lambda c.(\lambda a.([\![\text{D}_1]\!](a))(c))(\mathsf{Dwarf}) \\
&= \lambda c.([\![\text{D}_1]\!](\mathsf{Dwarf}))(c) \\
&= \lambda c.((\lambda P\lambda Q.\forall x.P(x) \to Q(x))(\mathsf{Dwarf}))(c) \\
&= \lambda c.(\lambda Q.\forall x.\mathsf{Dwarf}(x) \to Q(x))(c) \\
&= \lambda c.\forall x.\mathsf{Dwarf}(x) \to c(x)
\end{aligned}$$

Finally, we can compute $\overline{\text{S}} = \overline{\text{VP}}\texttt{<*>}\overline{\text{NP}_1}$ as follows:

$$
\begin{aligned}
\overline{\text{S}} &= \overline{\text{every dwarf loved some princess}} = \overline{\text{VP}_1}\texttt{<*>}\overline{\text{NP}_1} \\
&= \lambda k.\overline{\text{NP}_1}(\lambda n.\overline{\text{VP}}(\lambda m.k(m(n)))) \\
&= \lambda k.(\lambda c.\forall x.\mathsf{Dwarf}(x) \to c(x))(\lambda n.\overline{\text{VP}}(\lambda m.k(m(n)))) \\
&= \lambda k.\forall x.\mathsf{Dwarf}(x) \to (\lambda n.\overline{\text{VP}}(\lambda m.k(m(n))))(x) \\
&= \lambda k.\forall x.\mathsf{Dwarf}(x) \to \overline{\text{VP}}(\lambda m.k(m(x))) \\
&= \lambda k.\forall x.\mathsf{Dwarf}(x) \to (\lambda c.\exists y.\mathsf{Princess}(y) \wedge c(\mathsf{Love}(y)))(\lambda m.k(m(x))) \\
&= \lambda k.\forall x.\mathsf{Dwarf}(x) \to \exists y.\mathsf{Princess}(y) \wedge (\lambda m.k(m(x)))(\mathsf{Love}(y)) \\
&= \lambda k.\forall x.\mathsf{Dwarf}(x) \to \exists y.\mathsf{Princess}(y) \wedge k((\mathsf{Love}(y))(x)) \\
&= \lambda k.\forall x.\mathsf{Dwarf}(x) \to \exists y.\mathsf{Princess}(y) \wedge k(\mathsf{Love}(x, y))
\end{aligned}
$$

To get the meaning of S, we apply $\overline{\text{S}}$ to the trivial continuation $\lambda x.x$:

$$
\begin{aligned}
[\![\text{S}]\!] &= [\![\text{every dwarf loved some princess}]\!] = \overline{S}(\lambda x.x) \\
&= (\lambda k.\forall x.\mathsf{Dwarf}(x) \to \exists y.\mathsf{Princess}(y) \wedge k(\mathsf{Love}(x, y)))(\lambda x.x) \\
&= \forall x.\mathsf{Dwarf}(x) \to \exists y.\mathsf{Princess}(y) \wedge (\lambda x.x)(\mathsf{Love}(x, y)) \\
&= \forall x.\mathsf{Dwarf}(x) \to \exists y.\mathsf{Princess}(y) \wedge \mathsf{Love}(x, y)
\end{aligned}
$$

Note that with an alternative definition of $u\texttt{<*>'}v = \lambda k.u(\lambda m.v(\lambda n.k(m(n))))$[3], we can derive the reverse scope. The derivation is left as an exercise for the reader[4].

[3]van Eijck and Unger call this `cpsApply'`
[4]Specifically, Exercise 11.6