

Logic and Language: Exercise (Week 6)

Orestis Melkonian [6176208], Konstantinos Kogkalidis [6230067]

1 Syntax

1.1

First, we define the rules of *rightward extraction* $\hat{\alpha}_{\diamond}^r, \hat{\sigma}_{\diamond}^r$:

$$\frac{f : A \otimes (B \otimes \Diamond C) \rightarrow D}{\widehat{\alpha}_\Diamond^r f : (A \otimes B) \otimes \Diamond C \rightarrow D} \qquad \frac{f : (A \otimes \Diamond C) \otimes B \rightarrow D}{\widehat{\sigma}_\Diamond^r f : (A \otimes B) \otimes \Diamond C \rightarrow D}$$

We can now proceed with the derivation of

$$n \otimes ((n \setminus n) / (s / \diamond \square np)) \otimes ((np / n) \otimes n) \otimes ((np \setminus s) / np) \rightarrow n$$

as follows:

$$\begin{array}{c}
\frac{\overline{np \vdash np} \quad 1_{np} \quad \overline{n \vdash n} \quad 1_n}{\overline{np/n \vdash np/n}} \quad \backslash \\
\frac{\overline{np/n \vdash np/n}}{(np/n) \otimes n \vdash np} \quad \triangleright^{-1} \quad \frac{\overline{s \vdash s} \quad 1_s}{\overline{np \setminus s \vdash ((np/n) \otimes n) \setminus s}} \quad \backslash \quad \frac{\overline{np \vdash np} \quad 1_{np}}{\overline{\Box np \vdash \Box np}} \quad \Box \\
\frac{\overline{np \setminus s \vdash ((np/n) \otimes n) \setminus s}}{np \setminus s \vdash ((np/n) \otimes n) \setminus s} \quad \backslash \quad \frac{\overline{\Box np \vdash \Box np}}{\Diamond \Box np \vdash np} \quad \nabla^{-1} \\
\frac{\overline{np \setminus s \vdash ((np/n) \otimes n) \setminus s}}{(np \setminus s)/np \vdash (((np/n) \otimes n) \setminus s)/\Diamond \Box np} \quad / \\
\frac{(np \setminus s)/np \vdash (((np/n) \otimes n) \setminus s)/\Diamond \Box np}{(np \setminus s)/np \otimes \Diamond \Box np \vdash ((np/n) \otimes n) \setminus s} \quad \triangleright^{-1} \\
\frac{(np \setminus s)/np \otimes \Diamond \Box np \vdash ((np/n) \otimes n) \setminus s}{((np/n) \otimes n) \otimes ((np \setminus s)/np \otimes \Diamond \Box np) \vdash s} \quad \triangleleft^{-1} \\
\frac{((np/n) \otimes n) \otimes ((np \setminus s)/np \otimes \Diamond \Box np) \vdash s}{(((np/n) \otimes n) \otimes ((np \setminus s)/np)) \otimes \Diamond \Box np \vdash s} \quad \hat{\alpha}_{\Diamond}^r \\
\frac{(((np/n) \otimes n) \otimes ((np \setminus s)/np)) \otimes \Diamond \Box np \vdash s}{((np/n) \otimes n) \otimes ((np \setminus s)/np) \vdash s/\Diamond \Box np} \quad \triangleright \\
\frac{((np/n) \otimes n) \otimes ((np \setminus s)/np) \vdash s/\Diamond \Box np}{(n \setminus n)/(s/\Diamond \Box np) \vdash (n \setminus n)/(((np/n) \otimes n) \otimes (np \setminus s)/np)} \quad / \\
\frac{(n \setminus n)/(s/\Diamond \Box np) \vdash (n \setminus n)/(((np/n) \otimes n) \otimes (np \setminus s)/np)}{(n \setminus n)/(s/\Diamond \Box np) \otimes (((np/n) \otimes n) \otimes (np \setminus s)/np) \vdash n \setminus n} \quad \triangleright^{-1} \\
\frac{(n \setminus n)/(s/\Diamond \Box np) \otimes (((np/n) \otimes n) \otimes (np \setminus s)/np) \vdash n \setminus n}{n \otimes ((n \setminus n)/(s/\Diamond \Box np) \otimes (((np/n) \otimes n) \otimes (np \setminus s)/np)) \vdash n} \quad \triangleleft^{-1}
\end{array}$$

2 Interpretation

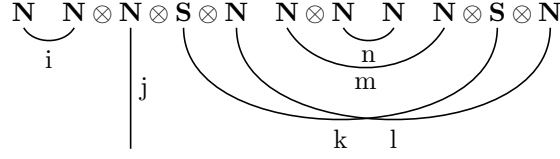
2.1

2.2

By working our way from the leaves of the proof tree, we get the following generalized Kronecker delta:

$$\begin{array}{l} \text{island}_i \otimes \text{that}_{j,k,l,m} \otimes \text{the}_{n,o} \otimes \text{hurricane}_p \otimes \text{destroyed}_{q,r,s} \xrightarrow{\delta_{j,t,r,s,q,p}^{i,k,l,m,n,o}} \mathbf{v}_r^{obj} \in \mathbf{N} \\ \mathbf{v}_r^{obj} = \text{island}_i \otimes \text{that}_{i,j,k,l} \otimes \text{the}_{m,n} \otimes \text{hurricane}_n \otimes \text{destroyed}_{m,k,l} \quad (\text{relabelled}) \end{array}$$

We give the matching diagram in the figure below:



2.3

In order to calculate the semantic value for the relative clause body 'the hurricane destroyed', we first apply **the**_{MN} to **hurricane**_N. The operation yields the noun-phrase **the hurricane**_M, represented by a row-vector equal to that of **hurricane**. The verb **destroyed**_{MKL} is then applied to the resulting vector, thereupon we obtain the final result **the hurricane destroyed**_{KL}. Concretely, **the hurricane destroyed**_{KL} = **destroyed**_{MKL}(**the**_{MN}**hurricane**_N) is a 2 by 3 matrix, the elements of which are:

$$\begin{pmatrix} 12 & -19 & 3 \\ 5 & 10 & 1 \end{pmatrix}$$

given by:

$$\text{the hurricane destroyed}(k, l) = \sum_{m \in M} \text{hurricane}(m) \times \text{destroyed}(m, k, l), \forall k \in K, l \in L$$

The corresponding Python code is given below:

```
import numpy as np
hurricane, island = np.array([3,-5,5]), np.array([-5,4,0])
the = np.eye(3)
destroyed = np.array([[4,-3,1],[5,5,2]], [[-1,-2,2],[2,-3,0]], [[-1,-4,2],[0,-4,-1]])
the_hurricane = np.matmul(the, hurricane) # == hurricane
the_hurricane_destroyed = np.tensordot(the_hurricane, destroyed, axes=1)
```

2.4

The interpreted type for the relative pronoun is:

$$[(n \setminus n) / (s / \diamond \square np)] = [n \setminus n] \otimes [s / \diamond \square np] = [n] \otimes [n] \otimes [s] \otimes [\diamond \square np] = N \otimes N \otimes S \otimes N$$

We can now give the following Frobenius recipe for **that**:

$$I \cong I \otimes I \xrightarrow{\eta_N \otimes \eta_N} N \otimes N \otimes N \otimes N \cong N \otimes N \otimes N \otimes I \otimes N \xrightarrow{1_N \otimes \mu_N \otimes \zeta_S \otimes 1_N} N \otimes N \otimes S \otimes N$$

In order to obtain the final interpretation, we do the following (dictated from the above recipe):

1. Reduce the rank of the transitive verb by summing over the S component, thus obtaining the following matrix:

$$\text{collapsed_destroyed} = \begin{pmatrix} \begin{pmatrix} 9 & 2 & 3 \end{pmatrix} \\ \begin{pmatrix} 1 & -5 & 2 \end{pmatrix} \\ \begin{pmatrix} -1 & -8 & 1 \end{pmatrix} \end{pmatrix}$$

2. Apply **collapsed_destroyed** to **the_hurricane** in subject position:

$$\mathbf{the_hurricane_destroyed} = \begin{pmatrix} 17 & -9 & 4 \end{pmatrix}$$

3. Multiply **the_hurricane_destroyed** element-wise with **island**:

$$\mathbf{island_that_the_hurricane_destroyed} = \begin{pmatrix} -85 & -36 & 0 \end{pmatrix}$$

The corresponding Python code is given below:

```
import numpy as np
hurricane, island = np.array([3,-5,5]), np.array([-5,4,0])
destroyed = np.array([[4,-3,1],[5,5,2]], [[-1,-2,2],[2,-3,0]], [[-1,-4,2],[0,-4,-1]])
the = np.eye(3)
the_hurricane = np.matmul(the, hurricane) # == hurricane
collapsed_destroyed = np.sum(destroyed, axis=1) # sum over the S dimension
the_hurricane_destroyed = np.matmul(hurricane, collapsed_destroyed)
island_that_the_hurricane_destroyed = island * hurricane_destroyed # element-wise
```
