Logic and Language: Exercise (Week 6)

Orestis Melkonian [6176208], Konstantinos Kogkalidis [6230067]

1 Syntax

1.1

First, we define the rules of *rightward extraction* $\widehat{\alpha}^r_\diamond$, $\widehat{\sigma}^r_\diamond$:

$$\frac{f : A \otimes (B \otimes \Diamond C) \to D}{\widehat{\alpha}^r_\diamond f : (A \otimes B) \otimes \Diamond C \to D} \qquad \frac{f : (A \otimes \Diamond C) \otimes B \to D}{\widehat{\sigma}^r_\diamond f : (A \otimes B) \otimes \Diamond C \to D}$$

We can now proceed with the derivation of

$$n \otimes ((n\backslash n)/(s/\Diamond\Box np)) \otimes ((np/n) \otimes n) \otimes ((np\backslash s)/np)) \to n$$

as follows:

$$\begin{prooftree}
\AxiomC{}\RightLabel{1_n}\UnaryInfC{$n \vdash n$}
\AxiomC{}\RightLabel{1_n}\UnaryInfC{$n \vdash n$}
\RightLabel{$\backslash$}\BinaryInfC{$n\backslash n \vdash n\backslash n$}
\AxiomC{}\RightLabel{1_{np}}\UnaryInfC{$np \vdash np$}
\AxiomC{}\RightLabel{1_n}\UnaryInfC{$n \vdash n$}
\RightLabel{$\backslash$}\BinaryInfC{$np\backslash n \vdash np\backslash n$}
\RightLabel{$\triangleright^{-1}$}\UnaryInfC{$(np\backslash n) \otimes n \vdash np$}
\AxiomC{}\RightLabel{1_s}\UnaryInfC{$s \vdash s$}
\RightLabel{$\backslash$}\BinaryInfC{$np\backslash s \vdash ((np\backslash n) \otimes n)\backslash s$}
\AxiomC{}\RightLabel{1_{np}}\UnaryInfC{$np \vdash np$}
\RightLabel{$\Box$}\UnaryInfC{$\Box np \vdash \Box np$}
\RightLabel{$\triangledown^{-1}$}\UnaryInfC{$\Diamond\Box np \vdash np$}
\RightLabel{$/$}\BinaryInfC{$(np\backslash s)/np \vdash (((np\backslash n) \otimes n)\backslash s)/\Diamond\Box np$}
\RightLabel{$\triangleright^{-1}$}\UnaryInfC{$(np\backslash s)/np \otimes \Diamond\Box np \vdash ((np\backslash n) \otimes n)\backslash s$}
\RightLabel{$\triangleleft^{-1}$}\UnaryInfC{$((np\backslash n) \otimes n) \otimes ((np\backslash s)/np \otimes \Diamond\Box np) \vdash s$}
\RightLabel{$\widehat{\alpha}^r_\diamond$}\UnaryInfC{$(((np\backslash n) \otimes n) \otimes ((np\backslash s)/np)) \otimes \Diamond\Box np \vdash s$}
\RightLabel{$\triangleright$}\UnaryInfC{$((np\backslash n) \otimes n) \otimes ((np\backslash s)/np) \vdash s/\Diamond\Box np$}
\RightLabel{$/$}\BinaryInfC{$(n\backslash n)/(s/\Diamond\Box np) \vdash (n\backslash n)/(((np\backslash n) \otimes n) \otimes (np\backslash s)/np)$}
\RightLabel{$\triangleright^{-1}$}\UnaryInfC{$(n\backslash n)/(s/\Diamond\Box np) \otimes (((np/n) \otimes (np\backslash s)/np) \vdash n\backslash n$}
\RightLabel{$\triangleleft^{-1}$}\UnaryInfC{$n \otimes ((n\backslash n)/(s/\Diamond\Box np) \otimes (((np/n) \otimes n) \otimes (np\backslash s)/np)) \vdash n$}
\end{prooftree}$$

2 Interpretation

2.1

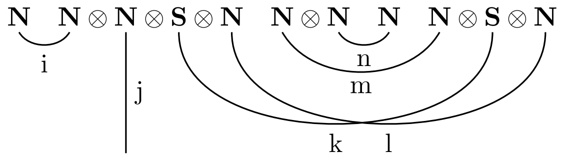
2.2

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

$$\mathbf{island}_i \otimes \mathbf{that}_{j,k,l,m} \otimes \mathbf{the}_{n,o} \otimes \mathbf{hurricane}_p \otimes \mathbf{destroyed}_{q,r,s} \xrightarrow{\delta^{i,k,l,m,n,o}_{j,t,r,s,q,p}} \mathbf{v}^{obj}_r \in \mathrm{N}$$

$$\mathbf{v}^{obj}_r = \mathbf{island}_i \otimes \mathbf{that}_{i,j,k,l} \otimes \mathbf{the}_{m,n} \otimes \mathbf{hurricane}_n \otimes \mathbf{destroyed}_{m,k,l} \quad \text{(relabeled)}$$

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 $\textbf{the}_{MN}$ to $\textbf{hurricane}_N$. The operation yields the noun-phrase $\textbf{the hurricane}_M$, represented by a row-vector equal to that of **hurricane**. The verb $\textbf{destroyed}_{MKL}$ is then applied to the resulting vector, thereupon we obtain the final result $\textbf{the hurricane destroyed}_{KL}$. Concretely, $\textbf{the hurricane destroyed}_{KL} = \textbf{destroyed}_{MKL}(\textbf{the}_{MN}\textbf{hurricane}_N)$ is a 2 by 3 matrix, the elements of which are:

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

given by:

$$\textbf{the hurricane destroyed}(k,l) = \sum_{n \epsilon N} \textbf{hurricane}(n) \times \textbf{destroyed}(n,k,l) \ \forall \ k \ \epsilon \ K, \ l \ \epsilon \ L$$

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.array([[1, 0,0], [0,1,0], [0,0,1]])
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:

$$\lceil (n\backslash n)/(s/\Diamond\Box np)\rceil = \lceil n\backslash n\rceil \otimes \lceil s/\Diamond\Box np\rceil = \lceil n\rceil \otimes \lceil n\rceil \otimes \lceil s\rceil \otimes \lceil \Diamond\Box np\rceil = N \otimes N \otimes S \otimes N$$

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:

$$\textbf{collapsed\_destroyed} = \begin{pmatrix} (\ 9 & 2 & 3\ ) \\ (\ 1 & -5 & 2\ ) \\ (\ -1 & -8 & 1\ ) \end{pmatrix}$$

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

$$\textbf{the\_hurricane\_destroyed} = \begin{pmatrix} 17 & -9 & 4 \end{pmatrix}$$

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

$$\textbf{island\_that\_the\_hurricane\_destroyed} = \begin{pmatrix} -85 & -36 & 0 \end{pmatrix}$$

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.array([[1, 0,0], [0,1,0], [0,0,1]])
the_hurricane = np.matmul(the, hurricane)  # == hurricane
collapsed_destroyed = destroyed[:,0,:] + destroyed[:,1,:]
the_hurricane_destroyed = np.matmul(hurricane, collapsed_destroyed)
island_that_the_hurricane_destroyed = island * hurricane_destroyed
```