Introduction to Glue Semantics

Class 2: Applications

Jamie Y. Findlay

University of Oslo

05 Aug 2025

Exercise 1: Resource sensitivity

1. *#Buffy was dead Willow last week.*
   Resource surplus: no consumer for *Willow*.
2. *#Willow finished the book and Tara put on the shelf.*
   Resource deficit: missing argument for *put*.
3. *#Xander attempted repeatedly.*
   Resource deficit: missing argument for *attempted*.
4. *#Giles has a grimoire which he has read it several times.*
   Resource surplus: no consumer for *it*.

Exercise 2: Modifying meaning constructors

With now needs to first combine with its mother rather than its `obj`:

$$\lambda y.\lambda x.\textbf{with}(x,y) : \boxed{\mathrm{E}(\hat{\bullet})} \multimap [E(\bullet) \multimap T(\bullet)]$$

And the meaning which the meaning constructor for `nmod` wants to first consume is related to its `case` dependent rather than its own node:

$$\lambda P.\lambda Q.\lambda x.P(x) \wedge Q(x) : \boxed{[\mathrm{E}(\bullet\ \texttt{case}) \multimap \mathrm{T}(\bullet\ \texttt{case})]} \multimap \\ [[E(\hat{\bullet}) \multimap T(\hat{\bullet})] \multimap E(\hat{\bullet}) \multimap T(\hat{\bullet})]$$

Exercise 3: Sentential adverbs

$$\dfrac{\begin{array}{l}[\textbf{\textit{unfortunately}}]\\ \lambda p.\mathbf{unfortunately}(p) :\\ T(d) \multimap T(d)\end{array} \qquad \dfrac{\begin{array}{l}[\textbf{\textit{Buffy}}]\\ \mathbf{buffy} :\\ E(b)\end{array} \qquad \begin{array}{l}[\textbf{\textit{died}}]\\ \lambda x.\mathbf{died}(x) :\\ E(b) \multimap T(d)\end{array}}{\begin{array}{l}\mathbf{died}(\mathbf{buffy}) :\\ T(d)\end{array}}}{\begin{array}{l}\mathbf{unfortunately}(\mathbf{died}(\mathbf{buffy})) :\\ T(d)\end{array}}$$

Exercise 4: Negation

$$\dfrac{\begin{array}{l}[\textbf{\textit{not}}]\\ \lambda p.\neg p :\\ T(a) \multimap T(a)\end{array} \qquad \dfrac{\begin{array}{l}[\textbf{\textit{Giles}}]\\ \textbf{giles} :\\ E(g)\end{array} \qquad \dfrac{\begin{array}{l}[\textbf{\textit{abandon}}]\\ \lambda y.\lambda x.\textbf{abandon}(x, y) :\\ E(b) \multimap [E(g) \multimap T(a)]\end{array} \quad \begin{array}{l}[\textbf{\textit{Buffy}}]\\ \textbf{buffy} :\\ E(b)\end{array}}{\begin{array}{l}\lambda x.\textbf{abandon}(x, \textbf{buffy}) :\\ E(g) \multimap T(a)\end{array}}}{\begin{array}{l}\textbf{abandon}(\textbf{giles}, \textbf{buffy}) :\\ T(a)\end{array}}}{\begin{array}{l}\neg\textbf{abandon}(\textbf{giles}, \textbf{buffy}) :\\ T(a)\end{array}}$$

Exercise 5: Control and raising (1/2)

(15) *Warren tried to shoot Buffy.*

$$\dfrac{\dfrac{\begin{array}{l}[\textit{Warren}]\\ \textbf{warren}:\\ E(w)\end{array} \quad \begin{array}{l}[\textit{tried}]\\ \lambda x.\lambda P.\textbf{tried}(x,P(x)):\\ E(w)\multimap[[E(w)\multimap T(s)]\multimap T(t)]\end{array}}{\begin{array}{l}\lambda P.\textbf{tried}(\textbf{warren},P(\textbf{warren})):\\ {[E(w)\multimap T(s)]\multimap T(t)}\end{array}} \quad \dfrac{\begin{array}{l}[\textit{shoot}]\\ \lambda y.\lambda x.\textbf{shoot}(x,y):\\ E(b)\multimap[E(w)\multimap T(s)]\end{array} \quad \begin{array}{l}[\textit{Buffy}]\\ \textbf{buffy}:\\ E(b)\end{array}}{\begin{array}{l}\lambda x.\textbf{shoot}(x,\textbf{buffy}):\\ E(w)\multimap T(s)\end{array}}}{\begin{array}{l}\textbf{tried}(\textbf{warren},\textbf{shoot}(\textbf{warren},\textbf{buffy})):\\ T(t)\end{array}}$$

Exercise 5: Control and raising (2/2)

(16) *Anya seems to love Xander.*

$$\dfrac{\begin{array}{l}[\textit{seems}]\\ \lambda p.\mathbf{seems}(p) : T(l) \multimap T(s)\end{array} \qquad \dfrac{\begin{array}{l}[\textit{Anya}]\\ \mathbf{anya} :\\ E(a)\end{array} \qquad \dfrac{\begin{array}{l}[\textit{love}]\\ \lambda y.\lambda x.\mathbf{love}(x,y) :\\ E(xa) \multimap [E(a) \multimap T(l)]\end{array} \quad \begin{array}{l}[\textit{Xander}]\\ \mathbf{xander} :\\ E(xa)\end{array}}{\begin{array}{l}\lambda x.\mathbf{love}(x,\mathbf{xander}) :\\ E(a) \multimap T(l)\end{array}}}{\begin{array}{l}\mathbf{love}(\mathbf{anya},\mathbf{xander}) :\\ T(l)\end{array}}}{\begin{array}{l}\mathbf{seems}(\mathbf{love}(\mathbf{anya},\mathbf{xander})) :\\ T(s)\end{array}}$$

Exercise 6: Putting it together (1/4)

1. *A red demon entered.*

$$
\dfrac{
  \dfrac{
    \begin{array}{l}
    [\mathit{a}]\\
    \lambda P.\lambda Q.\exists x.P(x) \wedge Q(x) :\\
    [E(d) \multimap T(d)] \multimap [[E(d) \multimap T(a)] \multimap T(a)]
    \end{array}
    \quad
    \dfrac{
      \dfrac{
        \begin{array}{l}
        [\texttt{amod}]\\
        \lambda P.\lambda Q.\lambda x.P(x) \wedge Q(x) :\\
        [E(r) \multimap T(r)] \multimap [[E(d) \multimap T(d)] \multimap E(d) \multimap T(d)]
        \end{array}
        \quad
        \begin{array}{l}
        [\mathit{red}]\\
        \lambda x.\mathbf{red}(x) :\\
        E(r) \multimap T(r)
        \end{array}
      }{
        \begin{array}{l}
        \lambda Q.\lambda x.\mathbf{red}(x) \wedge Q(x) :\\
        [E(d) \multimap T(d)] \multimap E(d) \multimap T(d)
        \end{array}
      }
      \quad
      \begin{array}{l}
      [\mathit{demon}]\\
      \lambda x.\mathbf{demon}(x) :\\
      E(d) \multimap T(d)
      \end{array}
    }{
      \begin{array}{l}
      \lambda x.\mathbf{red}(x) \wedge \mathbf{demon}(x) :\\
      E(d) \multimap T(d)
      \end{array}
    }
  }{
    \begin{array}{l}
    \lambda Q.\exists x.\mathbf{red}(x) \wedge \mathbf{demon}(x) \wedge Q(x) :\\
    [E(d) \multimap T(a)] \multimap T(a)
    \end{array}
  }
  \quad
  \begin{array}{l}
  [\mathit{appeared}]\\
  \lambda x.\mathbf{appeared}(x) :\\
  E(d) \multimap T(a)
  \end{array}
}{
  \begin{array}{l}
  \exists x.\mathbf{red}(x) \wedge \mathbf{demon}(x) \wedge \mathbf{appeared}(x) :\\
  T(a)
  \end{array}
}
$$

Exercise 6: Putting it together (2/4)

2. *Giles tried to read every book.*

$$
\dfrac{
  \dfrac{
    \begin{array}{l}\textbf{[}\textbf{\textit{Giles}}\textbf{]}\\ \textbf{giles} :\\ E(g)\end{array}
    \quad
    \begin{array}{l}\textbf{[}\textbf{\textit{tried}}\textbf{]}\\ \lambda x.\lambda P.\textbf{tried}(x, P(x)) :\\ E(g) \multimap [[E(g) \multimap T(r)] \multimap T(t)]\end{array}
  }{
    \begin{array}{l}\lambda P.\textbf{tried}(\textbf{giles}, P(\textbf{giles})) :\\ [E(g) \multimap T(r)] \multimap T(t)\end{array}
  }
  \quad
  \dfrac{
    \dfrac{
      \dfrac{
        \begin{array}{l}\textbf{[}\textbf{\textit{read}}\textbf{]}\\ \lambda x.\lambda y.\textbf{read}(x, y) :\\ E(g) \multimap [E(b) \multimap T(r)]\end{array}
        \quad
        \left[\begin{array}{l} x :\\ E(g)\end{array}\right]^{1}
      }{
        \begin{array}{l}\lambda y.\textbf{read}(x, y) :\\ E(b) \multimap T(r)\end{array}
      }
      \quad
      \begin{array}{l}\textbf{[}\textbf{\textit{every book}}\textbf{]}\\ \lambda Q.\forall y.\textbf{book}(y) \to \mathbf{Q}(y) :\\ [E(b) \multimap T(r)] \multimap T(r)\end{array}
    }{
      \begin{array}{l}\forall y.\textbf{book}(y) \to \textbf{read}(x, y) :\\ T(r)\end{array}
    }
  }{
    \begin{array}{l}\lambda x.\forall y.\textbf{book}(y) \to \textbf{read}(x, y) :\\ E(g) \multimap T(r)\end{array}
  }\multimap_{\mathcal{I},1}
}{
  \begin{array}{l}\textbf{tried}(\textbf{giles}, \forall y.\textbf{book}(y) \to \textbf{read}(\textbf{giles}, y)) :\\ T(t)\end{array}
}
$$

Exercise 6: Putting it together (3/4)

3. *Buffy slew every vampire that came.*

$$\dfrac{\dfrac{\begin{array}{l}[\textit{Buffy}]\\ \mathbf{buffy}:\\ E(b)\end{array}\quad\begin{array}{l}[\textit{slew}]\\ \lambda x.\lambda y.\mathbf{slew}(x,y):\\ E(b)\multimap[E(v)\multimap T(s)]\end{array}}{\begin{array}{l}\lambda y.\mathbf{slew}(\mathbf{buffy},y):\\ E(v)\multimap T(s)\end{array}}\quad\dfrac{\begin{array}{l}[\textit{every}]\\ \lambda P.\lambda Q.\forall x.P(x)\rightarrow Q(x):\\ [E(v)\multimap T(v)]\multimap\\ \quad[[E(v)\multimap T(s)]\multimap T(s)]\end{array}\quad\dfrac{\begin{array}{l}[\textit{vampire}]\\ \lambda x.\mathbf{vampire}(x):\\ E(v)\multimap T(v)\end{array}\quad\dfrac{\begin{array}{l}[\texttt{relcl}]\\ \lambda P.\lambda Q.\lambda x.Q(x)\wedge P(x):\\ [E(c\text{-}subj)\multimap T(c)]\multimap\\ \quad[E(v)\multimap T(v)]\multimap E(v)\multimap T(v)\end{array}\quad\begin{array}{l}[\textit{came}]\\ \lambda x.\mathbf{came}(x):\\ E(c\text{-}subj)\multimap T(c)\end{array}}{\begin{array}{l}\lambda Q.\lambda x.Q(x)\wedge\mathbf{came}(x):\\ [E(v)\multimap T(v)]\multimap E(v)\multimap T(v)\end{array}}}{\begin{array}{l}\lambda x.\mathbf{vampire}(x)\wedge\mathbf{came}(x):\\ E(v)\multimap T(v)\end{array}}}{\begin{array}{l}\lambda Q.\forall x.[\mathbf{vampire}(x)\wedge\mathbf{came}(x)]\rightarrow Q(x):\\ [E(v)\multimap T(s)]\multimap T(s)\end{array}}}{\begin{array}{l}\forall x.[\mathbf{vampire}(x)\wedge\mathbf{came}(x)]\rightarrow\mathbf{slew}(\mathbf{buffy},x):\\ T(s)\end{array}}$$

Exercise 6: Putting it together (4/4)

4. *Tara did not seem to suffer*

$$\dfrac{\begin{array}{l}\textbf{[}\textbf{\textit{not}}\textbf{]}\\ \lambda p.\neg p :\\ T(se) \multimap T(se)\end{array} \qquad \dfrac{\begin{array}{l}\textbf{[}\textbf{\textit{seem}}\textbf{]}\\ \lambda p.\mathbf{seem}\big(p\big) :\\ T(su) \multimap T(se)\end{array} \qquad \dfrac{\begin{array}{l}\textbf{[}\textbf{\textit{Tara}}\textbf{]}\\ \mathbf{tara} :\\ E(t)\end{array} \quad \begin{array}{l}\textbf{[}\textbf{\textit{suffer}}\textbf{]}\\ \lambda x.\mathbf{suffer}(x) :\\ E(t) \multimap T(su)\end{array}}{\begin{array}{l}\mathbf{suffer(tara)} :\\ T(su)\end{array}}}{\begin{array}{l}\mathbf{seem}\big(\mathbf{suffer(tara)}\big) :\\ T(se)\end{array}}}{\begin{array}{l}\neg\mathbf{seem}\big(\mathbf{suffer(tara)}\big) :\\ T(se)\end{array}}$$

For next time

- Starting tomorrow, Mark will guide you through a computational implementation of Glue.
- Before then, it would be good if you could try to download and install the requisite software (but don't worry if you can't get it working – we will be there to help troubleshoot tomorrow, or you can ask on the Discord).

Installing the computational Glue system (1/2)

1. Install Docker:
   https://www.docker.com/products/docker-desktop/
   (the free version is completely sufficient).
2. Download the UD+Glue project from here:
   https://github.com/Mmaz1988/xleplusglue/tree/ESSLLI2025_glue_course
   (click `code` > `download ZIP`)
3. Unzip the project somewhere you will be able to find again.
4. The project contains a Stanza folder. Setup the Stanza folder as described in the `README.md` within the Stanza folder.
   - **IMPORTANT:** The Stanza folder contains `.sh` files that you can execute to automate the process (via the terminal/other command line tool; more info in the `README.md`). There is one file for Windows users and one file for Unix users (which should run on both macOS and Linux).

Installing the computational Glue system (2/2)

5. Start Docker (by starting the Docker Desktop app).
6. Navigate to the `Docker` folder within the project folder.
7. Open a terminal in that location and run the following command: `docker compose up --build`. Leave the terminal open while the installation and setup process finishes. This might take a little while, depending on your Internet connection.
8. Once this is done, you should be able to open UD+Glue by launching a browser and navigating to localhost:80.
9. There, you should be able to navigate to analysis (via the pop-in menu on the left) and analyze the default sentence. You should also be able to run the testsuite in regression testing by clicking the “Parse all” button under the settings.
   (Trying these steps just makes sure everything is installed correctly. We will talk through actually using the tools tomorrow!)