LING3804: Lecture Notes 10
Compositional Semantics

How do we get semantic representations from natural language sentences?

We can assemble logical expressions from adjacent phrases and clauses using beta reduction.

Contents

10.1 Composition rules

We define composition rules for functions φ and ψ with types γ and δ:

$$\psi : \delta, \quad \varphi : \gamma\textbf{-a}\delta \;\Rightarrow\; (\,\varphi\,\psi\,) : \gamma \qquad \text{application ahead (Aa)}$$
$$\varphi : \gamma\textbf{-b}\delta, \quad \psi : \delta \;\Rightarrow\; (\,\varphi\,\psi\,) : \gamma \qquad \text{application behind (Ab)}$$

For example, if we have the following very simple meanings for *Kim* and *sleeps*:

$$\textit{Kim} \Rightarrow \mathsf{Person43} : \mathbf{N}$$
$$\textit{sleeps} \Rightarrow \mathsf{Sleep} : \mathbf{V\text{-}aN}$$

we can combine them like this:

$$\mathsf{Person43} : \mathbf{N}, \;\; \mathsf{Sleep} : \mathbf{V\text{-}aN} \;\Rightarrow\; (\mathsf{Sleep}\ \mathsf{Person43}) : \mathbf{V} \qquad \text{(Aa)}$$

We also define reductions for sets ρ and σ, entity terms χ and ω, and statements φ:

$$\mathsf{Some}\ (\lambda_\chi\ \mathsf{Equal}\ \omega\,\chi \wedge \rho\,\chi)\ \sigma \;=\; \rho\,\omega \wedge \sigma\,\omega \qquad \text{equality elimination ahead (EEa)}$$
$$\mathsf{Some}\ (\lambda_\chi\ \rho\,\chi \wedge \mathsf{Equal}\ \omega\,\chi)\ \sigma \;=\; \rho\,\omega \wedge \sigma\,\omega \qquad \text{equality elimination behind (EEb)}$$
$$\mathsf{True} \wedge \varphi \;=\; \varphi \qquad \text{tautology elimination ahead (TEa)}$$
$$\varphi \wedge \mathsf{True} \;=\; \varphi \qquad \text{tautology elimination behind (TEb)}$$

For example:

$$\mathsf{Some}\ (\lambda_x\ \mathsf{Equal}\ \mathsf{Kim}\ x \wedge \mathsf{Prof}\ x)\ (\lambda_x\ \mathsf{Sleep}\ x) \;\Rightarrow\; \mathsf{Prof}\ \mathsf{Kim} \wedge \mathsf{Sleep}\ \mathsf{Kim} \qquad \text{(EEa)}$$

Practice 10.1:

Simplify the following:

$$\textsf{Prof Kim} \wedge \textsf{True}$$

10.2 Example composition: content questions

We extend this system to compose meanings for questions by treating noun phrases as quantifiers.

Given the following categories:

$$\begin{aligned}
\mathbf{W} &: \text{wh-question} \\
\mathbf{WBAR} &: \text{wh-phrase} \\
\mathbf{N} &: \text{noun phrase} \\
\mathbf{NBAR} &: \text{common noun}
\end{aligned}$$

and the following lexical entries, treating each question as a desire (Want) to know (Believe):

$$\begin{array}{l}
\textit{who} \Rightarrow (\lambda_r\ (\lambda_s\ \textsf{All}\ (\lambda_i\ \textsf{CurrentTheory}\ i) \\
\qquad\qquad (\lambda_i\ \textsf{All}\ (\lambda_t\ \textsf{CurrentTimeInTheory}\ t\ i) \\
\qquad\qquad\quad (\lambda_t\ \textsf{All}\ (\lambda_a\ \textsf{Speaker}\ a\ t) \\
\qquad\qquad\qquad (\lambda_a\ \textsf{All}\ (\lambda_x\ r\ x \wedge s\ x) \\
\qquad\qquad\qquad\quad (\lambda_x\ \textsf{All}\ (\lambda_j\ \textsf{Equal}\ j\ (\uparrow r\ x \wedge s\ x)) \\
\qquad\qquad\qquad\qquad (\lambda_j\ \textsf{All}\ (\lambda_k\ \textsf{Equal}\ k\ (\uparrow \textsf{All}\ (\lambda_u\ \textsf{CurrentTimeInTheory}\ k\ u) \\
\qquad\qquad\qquad\qquad\qquad (\lambda_u\ \textsf{All}\ (\lambda_v\ \textsf{ConsecutiveTo}\ u\ v) \\
\qquad\qquad\qquad\qquad\qquad\quad (\lambda_v\ \textsf{Believe}\ j\ a\ v)))) \\
\qquad\qquad\qquad\qquad\quad (\lambda_k\ \textsf{Want}\ k\ a\ t)))))))))) : \mathbf{WBAR} \\
\textit{is} \Rightarrow (\lambda_p\ (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \textsf{True})\ (\lambda_x\ p\ (\lambda_y\ \textsf{Equal}\ x\ y)\ (\lambda_y\ \textsf{True})))))) : \mathbf{W\text{-}aWBAR\text{-}bN} \\
\textit{a} \Rightarrow (\lambda_q\ (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_z\ r\ z \wedge q\ (\lambda_x\ \textsf{Equal}\ z\ x)\ (\lambda_z\ \textsf{True}))\ s))) : \mathbf{N\text{-}bNBAR} \\
\textit{student} \Rightarrow (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_y\ \textsf{Student}\ y \wedge r\ y)\ s)) : \mathbf{NBAR}
\end{array}$$

we can derive the logical form for *Who is a student?* shown in the previous lecture notes.

This involves treating sentence words as premises and combining them according to their syntax:

$$\begin{array}{lll}
\textit{who, is, a, student} \Rightarrow \textit{who, is,} & (\lambda_q\ (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_z\ r\ z \wedge q\ (\lambda_x\ \textsf{Equal}\ z\ x)\ (\lambda_x\ \textsf{True}))\ s))) : \mathbf{N\text{-}bNBAR}, & \text{def} \\
 & (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_y\ \textsf{Student}\ y \wedge r\ y)\ s)) : \mathbf{NBAR} & \\
\Rightarrow \textit{who, is,} & (\lambda_q\ (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_z\ r\ z \wedge q\ (\lambda_x\ \textsf{Equal}\ z\ x)\ (\lambda_x\ \textsf{True}))\ s))) & \text{Ab} \\
 & (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_y\ \textsf{Student}\ y \wedge r\ y)\ s)) : \mathbf{N} & \\
= \textit{who, is,} & (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_z\ r\ z \wedge\ (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_y\ \textsf{Student}\ y \wedge r\ y)\ s)) & \text{BR on } q \\
 & \qquad (\lambda_x\ \textsf{Equal}\ z\ x) & \\
 & \qquad (\lambda_x\ \textsf{True}))\ s)) : \mathbf{N} & \\
= \textit{who, is,} & (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_z\ r\ z \wedge\ (\lambda_s\ \textsf{Some}\ (\lambda_y\ \textsf{Student}\ y \wedge (\lambda_x\ \textsf{Equal}\ z\ x)\ y)\ s)) & \text{BR on } r \\
 & \qquad (\lambda_x\ \textsf{True})\ s)) : \mathbf{N} &
\end{array}$$

$$= \mathit{who}, \mathit{is}, \ (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge\ (\lambda_s\ \mathsf{Some}\ (\lambda_y\ \mathsf{Student}\ y \wedge \mathsf{Equal}\ z\ y)\ s)) \ (\lambda_x\ \mathsf{True})\ s)) : \mathbf{N} \qquad \text{BR on } x$$

$$= \mathit{who}, \mathit{is}, \ (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge \mathsf{Some}\ \ (\lambda_y\ \mathsf{Student}\ y \wedge \mathsf{Equal}\ z\ y) \ (\lambda_x\ \mathsf{True}))\ s)) : \mathbf{N} \qquad \text{BR on } s$$

$$= \mathit{who}, \mathit{is}, (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge \mathsf{Student}\ z \wedge (\lambda_x \mathsf{True})\ z)\ s)) : \mathbf{N} \qquad \text{EE}$$

$$= \mathit{who}, \mathit{is}, (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge \mathsf{Student}\ z \wedge \mathsf{True})\ s)) : \mathbf{N} \qquad \text{BR on } x$$

$$= \mathit{who}, \mathit{is}, (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge \mathsf{Student}\ z)\ s)) : \mathbf{N} \qquad \text{TE}$$

$$\Rightarrow \mathit{who}, \ (\lambda_p\ (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ p\ \ (\lambda_y\ \mathsf{Equal}\ x\ y) \ (\lambda_y\ \mathsf{True})))))) : \mathbf{W\text{-}aWBAR\text{-}bN}, \ (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge \mathsf{Student}\ z)\ s)) : \mathbf{N} \qquad \text{def}$$

$$\Rightarrow \mathit{who}, \ (\lambda_p\ (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ p\ (\lambda_y\ \mathsf{Equal}\ x\ y)\ (\lambda_y\ \mathsf{True}))))) \ (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge \mathsf{Student}\ z)\ s)) : \mathbf{W\text{-}aWBAR} \qquad \text{Ab}$$

$$= \mathit{who}, (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \ (\lambda_r\ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ r\ z \wedge \mathsf{Student}\ z)\ s)) \ (\lambda_y\ \mathsf{Equal}\ x\ y) \ (\lambda_y\ \mathsf{True}))))) : \mathbf{W\text{-}aWBAR} \qquad \text{BR on } p$$

$$= \mathit{who}, (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ (\lambda_y\ \mathsf{Equal}\ x\ y)\ z \wedge \mathsf{Student}\ z)\ s) \ (\lambda_y\ \mathsf{True}))))) : \mathbf{W\text{-}aWBAR} \qquad \text{BR on } r$$

$$= \mathit{who}, (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \ (\lambda_s\ \mathsf{Some}\ (\lambda_z\ \mathsf{Equal}\ x\ z \wedge \mathsf{Student}\ z)\ s) \ (\lambda_y\ \mathsf{True}))))) : \mathbf{W\text{-}aWBAR} \qquad \text{BR on } y$$

$$= \mathit{who}, (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \mathsf{Some}\ \ (\lambda_z\ \mathsf{Equal}\ x\ z \wedge \mathsf{Student}\ z) \ (\lambda_y\ \mathsf{True}))))) : \mathbf{W\text{-}aWBAR} \qquad \text{BR on } s$$

$$= \mathit{who}, (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \mathsf{Student}\ x \wedge (\lambda_y\ \mathsf{True})\ x)))) : \mathbf{W\text{-}aWBAR} \qquad \text{EE}$$

$$= \mathit{who}, (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \mathsf{Student}\ x \wedge \mathsf{True})))) : \mathbf{W\text{-}aWBAR} \qquad \text{BR on } y$$

$$= \mathit{who}, (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \mathsf{Student}\ x)))) : \mathbf{W\text{-}aWBAR} \qquad \text{TE}$$

$$\begin{aligned} = \ & (\lambda_r\ (\lambda_s\ \mathsf{All}\ \ (\lambda_i\ \mathsf{CurrentTheory}\ i) \\ & \quad (\lambda_i\ \mathsf{All}\ \ (\lambda_t\ \mathsf{CurrentTimeInTheory}\ t\ i) \\ & \quad\quad (\lambda_t\ \mathsf{All}\ \ (\lambda_a\ \mathsf{Speaker}\ a\ t) \\ & \quad\quad\quad (\lambda_a\ \mathsf{All}\ \ (\lambda_x\ r\ x \wedge s\ x) \\ & \quad\quad\quad\quad (\lambda_x\ \mathsf{All}\ \ (\lambda_j\ \mathsf{Equal}\ j\ (\uparrow r\ x \wedge s\ x)) \\ & \quad\quad\quad\quad\quad (\lambda_j\ \mathsf{All}\ \ (\lambda_k\ \mathsf{Equal}\ k\ (\uparrow \ldots)) \\ & \quad\quad\quad\quad\quad\quad (\lambda_k\ \mathsf{Want}\ k\ a\ t)))))))))) : \mathbf{WBAR}, \\ & (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \mathsf{Student}\ x)))) : \mathbf{W\text{-}aWBAR} \end{aligned} \qquad \text{def}$$

$$\begin{aligned} = \ & (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \mathsf{True})\ (\lambda_x\ \mathsf{Student}\ x)))) \\ & (\lambda_r\ (\lambda_s\ \mathsf{All}\ \ (\lambda_i\ \mathsf{CurrentTheory}\ i) \\ & \quad (\lambda_i\ \mathsf{All}\ \ (\lambda_t\ \mathsf{CurrentTimeInTheory}\ t\ i) \\ & \quad\quad (\lambda_t\ \mathsf{All}\ \ (\lambda_a\ \mathsf{Speaker}\ a\ t) \\ & \quad\quad\quad (\lambda_a\ \mathsf{All}\ \ (\lambda_x\ r\ x \wedge s\ x) \\ & \quad\quad\quad\quad (\lambda_x\ \mathsf{All}\ \ (\lambda_j\ \mathsf{Equal}\ j\ (\uparrow r\ x \wedge s\ x)) \\ & \quad\quad\quad\quad\quad (\lambda_j\ \mathsf{All}\ \ (\lambda_k\ \mathsf{Equal}\ k\ (\uparrow \ldots)) \\ & \quad\quad\quad\quad\quad\quad (\lambda_k\ \mathsf{Want}\ k\ a\ t)))))))))) : \mathbf{W} \end{aligned} \qquad \text{Aa}$$

$$
\begin{array}{ll}
= (\lambda_r\ (\lambda_s\ (\lambda_r\ (\lambda_s\ \textsf{All}\ (\lambda_i\ \textsf{CurrentTheory}\ i) & \text{BR on } q \\
\qquad\qquad (\lambda_i\ \textsf{All}\ (\lambda_t\ \textsf{CurrentTimeInTheory}\ t\ i) & \\
\qquad\qquad\quad (\lambda_t\ \textsf{All}\ (\lambda_a\ \textsf{Speaker}\ a\ t) & \\
\qquad\qquad\qquad (\lambda_a\ \textsf{All}\ (\lambda_x\ r\ x \wedge s\ x) & \\
\qquad\qquad\qquad\quad (\lambda_x\ \textsf{All}\ (\lambda_j\ \textsf{Equal}\ j\ (\uparrow r\ x \wedge s\ x)) & \\
\qquad\qquad\qquad\qquad (\lambda_j\ \textsf{All}\ (\lambda_k\ \textsf{Equal}\ k\ (\uparrow \ldots)) & \\
\qquad\qquad\qquad\qquad\quad (\lambda_k\ \textsf{Want}\ k\ a\ t)))))))) & \\
\quad (\lambda_x\ \textsf{True}) & \\
\quad (\lambda_x\ \textsf{Student}\ x))) : \mathbf{W} &
\end{array}
$$

$$
\begin{array}{ll}
= (\lambda_r\ (\lambda_s\ (\lambda_s\ \textsf{All}\ (\lambda_i\ \textsf{CurrentTheory}\ i) & \text{BR on } r \\
\qquad\qquad (\lambda_i\ \textsf{All}\ (\lambda_t\ \textsf{CurrentTimeInTheory}\ t\ i) & \\
\qquad\qquad\quad (\lambda_t\ \textsf{All}\ (\lambda_a\ \textsf{Speaker}\ a\ t) & \\
\qquad\qquad\qquad (\lambda_a\ \textsf{All}\ (\lambda_x\ (\lambda_x\ \textsf{True})\ x \wedge s\ x) & \\
\qquad\qquad\qquad\quad (\lambda_x\ \textsf{All}\ (\lambda_j\ \textsf{Equal}\ j\ (\uparrow (\lambda_x\ \textsf{True})\ x \wedge s\ x)) & \\
\qquad\qquad\qquad\qquad (\lambda_j\ \textsf{All}\ (\lambda_k\ \textsf{Equal}\ k\ (\uparrow \ldots)) & \\
\qquad\qquad\qquad\qquad\quad (\lambda_k\ \textsf{Want}\ k\ a\ t)))))))) & \\
\quad (\lambda_x\ \textsf{Student}\ x))) : \mathbf{W} &
\end{array}
$$

$$
\begin{array}{ll}
= (\lambda_r\ (\lambda_s\ (\lambda_s\ \textsf{All}\ (\lambda_i\ \textsf{CurrentTheory}\ i) & \text{BR on } x \\
\qquad\qquad (\lambda_i\ \textsf{All}\ (\lambda_t\ \textsf{CurrentTimeInTheory}\ t\ i) & \\
\qquad\qquad\quad (\lambda_t\ \textsf{All}\ (\lambda_a\ \textsf{Speaker}\ a\ t) & \\
\qquad\qquad\qquad (\lambda_a\ \textsf{All}\ (\lambda_x\ \textsf{True} \wedge s\ x) & \\
\qquad\qquad\qquad\quad (\lambda_x\ \textsf{All}\ (\lambda_j\ \textsf{Equal}\ j\ (\uparrow \textsf{True} \wedge s\ x)) & \\
\qquad\qquad\qquad\qquad (\lambda_j\ \textsf{All}\ (\lambda_k\ \textsf{Equal}\ k\ (\uparrow \ldots)) & \\
\qquad\qquad\qquad\qquad\quad (\lambda_k\ \textsf{Want}\ k\ a\ t)))))))) & \\
\quad (\lambda_x\ \textsf{Student}\ x))) : \mathbf{W} &
\end{array}
$$

$$
\begin{array}{ll}
= (\lambda_r\ (\lambda_s\ (\lambda_s\ \textsf{All}\ (\lambda_i\ \textsf{CurrentTheory}\ i) & \text{TE} \\
\qquad\qquad (\lambda_i\ \textsf{All}\ (\lambda_t\ \textsf{CurrentTimeInTheory}\ t\ i) & \\
\qquad\qquad\quad (\lambda_t\ \textsf{All}\ (\lambda_a\ \textsf{Speaker}\ a\ t) & \\
\qquad\qquad\qquad (\lambda_a\ \textsf{All}\ (\lambda_x\ s\ x) & \\
\qquad\qquad\qquad\quad (\lambda_x\ \textsf{All}\ (\lambda_j\ \textsf{Equal}\ j\ (\uparrow s\ x)) & \\
\qquad\qquad\qquad\qquad (\lambda_j\ \textsf{All}\ (\lambda_k\ \textsf{Equal}\ k\ (\uparrow \ldots)) & \\
\qquad\qquad\qquad\qquad\quad (\lambda_k\ \textsf{Want}\ k\ a\ t)))))))) & \\
\quad (\lambda_x\ \textsf{Student}\ x))) : \mathbf{W} &
\end{array}
$$

$$
\begin{array}{ll}
= (\lambda_r\ (\lambda_s\ \textsf{All}\ (\lambda_i\ \textsf{CurrentTheory}\ i) & \text{BR on } s \\
\qquad\qquad (\lambda_i\ \textsf{All}\ (\lambda_t\ \textsf{CurrentTimeInTheory}\ t\ i) & \\
\qquad\qquad\quad (\lambda_t\ \textsf{All}\ (\lambda_a\ \textsf{Speaker}\ a\ t) & \\
\qquad\qquad\qquad (\lambda_a\ \textsf{All}\ (\lambda_x\ (\lambda_x\ \textsf{Student}\ x)\ x) & \\
\qquad\qquad\qquad\quad (\lambda_x\ \textsf{All}\ (\lambda_j\ \textsf{Equal}\ j\ (\uparrow (\lambda_x\ \textsf{Student}\ x)\ x)) & \\
\qquad\qquad\qquad\qquad (\lambda_j\ \textsf{All}\ (\lambda_k\ \textsf{Equal}\ k\ (\uparrow \ldots)) & \\
\qquad\qquad\qquad\qquad\quad (\lambda_k\ \textsf{Want}\ k\ a\ t)))))))) : \mathbf{W} &
\end{array}
$$

$$
\begin{array}{l}
= (\lambda_r\ (\lambda_s\ \mathsf{All}\ \ (\lambda_i\ \mathsf{CurrentTheory}\ i) \\
\qquad (\lambda_i\ \mathsf{All}\ \ (\lambda_t\ \mathsf{CurrentTimeInTheory}\ t\ i) \\
\qquad\quad (\lambda_t\ \mathsf{All}\ \ (\lambda_a\ \mathsf{Speaker}\ a\ t) \\
\qquad\qquad (\lambda_a\ \mathsf{All}\ \ (\lambda_x\ \mathsf{Student}\ x) \\
\qquad\qquad\quad (\lambda_x\ \mathsf{All}\ \ (\lambda_j\ \mathsf{Equal}\ j\ (\uparrow\ \mathsf{Student}\ x)) \\
\qquad\qquad\qquad (\lambda_j\ \mathsf{All}\ \ (\lambda_k\ \mathsf{Equal}\ k\ (\uparrow\ \ldots)) \\
\qquad\qquad\qquad\quad (\lambda_k\ \mathsf{Want}\ k\ a\ t)))))))) : \mathbf{W}
\end{array}
$$

BR on x

We then provide unconstrained arguments for the (here unused) restrictor and nuclear scope:

$$
\begin{array}{l}
(\lambda_r\ (\lambda_s\ \mathsf{All}\ \ (\lambda_i\ \mathsf{CurrentTheory}\ i) \\
\qquad (\lambda_i\ \mathsf{All}\ \ (\lambda_t\ \mathsf{CurrentTimeInTheory}\ t\ i) \\
\qquad\quad (\lambda_t\ \mathsf{All}\ \ (\lambda_a\ \mathsf{Speaker}\ a\ t) \\
\qquad\qquad (\lambda_a\ \mathsf{All}\ \ (\lambda_x\ \mathsf{Student}\ x) \\
\qquad\qquad\quad (\lambda_x\ \mathsf{All}\ \ (\lambda_j\ \mathsf{Equal}\ j\ (\uparrow\ \mathsf{Student}\ x)) \\
\qquad\qquad\qquad (\lambda_j\ \mathsf{All}\ \ (\lambda_k\ \mathsf{Equal}\ k\ (\uparrow\ \mathsf{All}\ \ (\lambda_u\ \mathsf{CurrentTimeInTheory}\ k\ u) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\lambda_u\ \mathsf{All}\ \ (\lambda_v\ \mathsf{ConsecutiveTo}\ u\ v) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (\lambda_v\ \mathsf{Believe}\ \ j\ a\ v)))) \\
\qquad\qquad\qquad\quad (\lambda_k\ \mathsf{Want}\ k\ a\ t)))))))) : \mathbf{W} \\
(\lambda_z \mathsf{True}) \\
(\lambda_z \mathsf{True})
\end{array}
$$

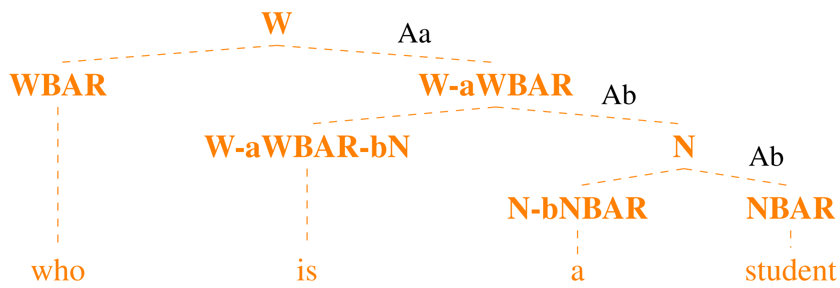
$$
\begin{array}{l}
=\ \ (\lambda_s\ \mathsf{All}\ \ (\lambda_i\ \mathsf{CurrentTheory}\ i) \\
\qquad (\lambda_i\ \mathsf{All}\ \ (\lambda_t\ \mathsf{CurrentTimeInTheory}\ t\ i) \\
\qquad\quad (\lambda_t\ \mathsf{All}\ \ (\lambda_a\ \mathsf{Speaker}\ a\ t) \\
\qquad\qquad (\lambda_a\ \mathsf{All}\ \ (\lambda_x\ \mathsf{Student}\ x) \\
\qquad\qquad\quad (\lambda_x\ \mathsf{All}\ \ (\lambda_j\ \mathsf{Equal}\ j\ (\uparrow\ \mathsf{Student}\ x)) \\
\qquad\qquad\qquad (\lambda_j\ \mathsf{All}\ \ (\lambda_k\ \mathsf{Equal}\ k\ (\uparrow\ \mathsf{All}\ \ (\lambda_u\ \mathsf{CurrentTimeInTheory}\ k\ u) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\lambda_u\ \mathsf{All}\ \ (\lambda_v\ \mathsf{ConsecutiveTo}\ u\ v) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (\lambda_v\ \mathsf{Believe}\ \ j\ a\ v)))) \\
\qquad\qquad\qquad\quad (\lambda_k\ \mathsf{Want}\ k\ a\ t))))))) : \mathbf{W} \\
\quad (\lambda_z \mathsf{True})
\end{array}
$$

BR on r

$$
\begin{array}{l}
= \mathsf{All}\ \ (\lambda_i\ \mathsf{CurrentTheory}\ i) \\
\qquad (\lambda_i\ \mathsf{All}\ \ (\lambda_t\ \mathsf{CurrentTimeInTheory}\ t\ i) \\
\qquad\quad (\lambda_t\ \mathsf{All}\ \ (\lambda_a\ \mathsf{Speaker}\ a\ t) \\
\qquad\qquad (\lambda_a\ \mathsf{All}\ \ (\lambda_x\ \mathsf{Student}\ x) \\
\qquad\qquad\quad (\lambda_x\ \mathsf{All}\ \ (\lambda_j\ \mathsf{Equal}\ j\ (\uparrow\ \mathsf{Student}\ x)) \\
\qquad\qquad\qquad (\lambda_j\ \mathsf{All}\ \ (\lambda_k\ \mathsf{Equal}\ k\ (\uparrow\ \mathsf{All}\ \ (\lambda_u\ \mathsf{CurrentTimeInTheory}\ k\ u) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\lambda_u\ \mathsf{All}\ \ (\lambda_v\ \mathsf{ConsecutiveTo}\ u\ v) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (\lambda_v\ \mathsf{Believe}\ \ j\ a\ v)))) \\
\qquad\qquad\qquad\quad (\lambda_k\ \mathsf{Want}\ k\ a\ t)))))) : \mathbf{W}
\end{array}
$$

BR on s

The above derivation corresponds to the following tree:



Practice 10.2:

Using the above composition rules and the below lexical entries:

$$\textit{everyone} \Rightarrow (\lambda_r\ (\lambda_s\ \textsf{All}\ (\lambda_x\ \textsf{Person}\ x \wedge r\ x)\ s)) : \textbf{N}$$
$$\textit{likes} \Rightarrow (\lambda_p\ (\lambda_q\ (\lambda_r\ (\lambda_s\ q\ (\lambda_x\ \textsf{True})\ (\lambda_x\ p\ (\lambda_y\ \textsf{True})\ (\lambda_y\ \textsf{Like}\ y\ x)))) : \textbf{V-aN-bN}$$
$$\textit{something} \Rightarrow (\lambda_r\ (\lambda_s\ \textsf{Some}\ (\lambda_y\ \textsf{Thing}\ y \wedge r\ y)\ s)) : \textbf{N}$$

do the following:

1. draw a syntax tree for the sentence *everyone likes something*,
2. derive a logical form for this sentence based on your syntax tree.

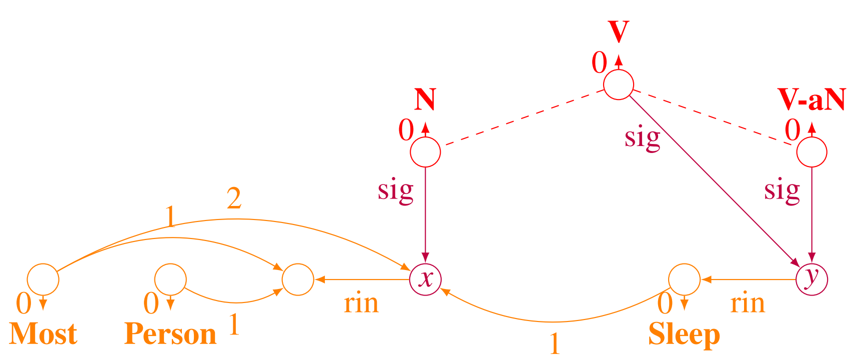
10.3 Extra: Composition with cued associations

We can also define this kind of assembly using cued associations in associative memory.

10.3.1 Signs (de Saussure, 1916)

We've seen how logical expressions can be represented and reasoned about as cued associations.

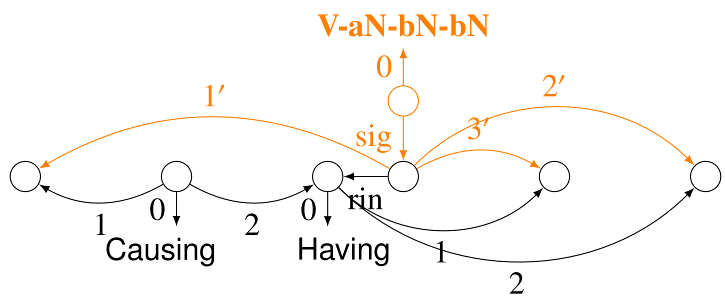
We can add these to lexical entries to form **signs**, which **signify** events:



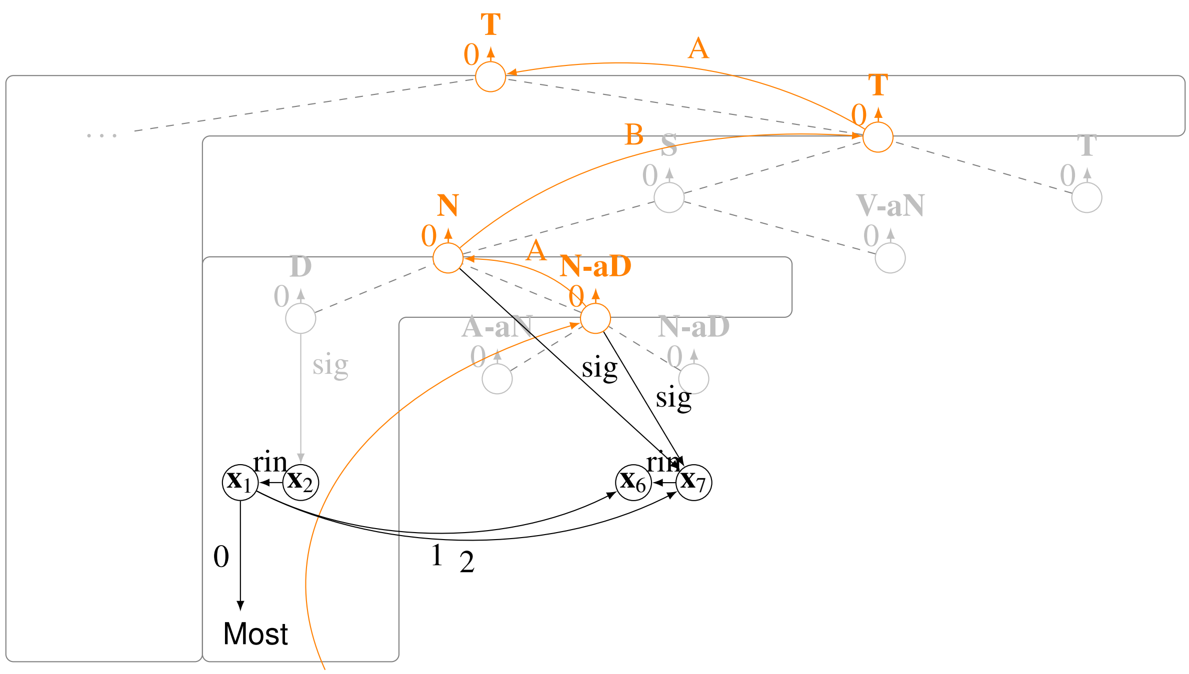
Signs have:

- **signified** structures (edges labeled sig) – these are our complex ideas;
- **syntactic categories** (edges labeled 0) – we've seen these already (**V**, **V-aN**, etc.);
- **syntactic arguments** (labeled 1′, 2′, etc., from signified), connecting semantic participants;
- **inheritance** associations (labeled rin), to make restrictions accessible from nuclear scope.
- **apex/base** associations (labeled A, and B), connecting derivation fragments on the store;

For example, here's a lexical sign for the word *give*, defined to mean *cause to have*:



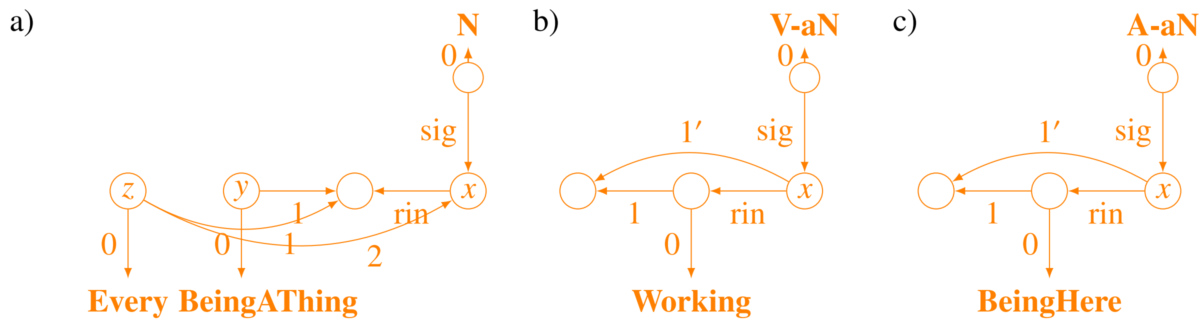
And here's a store of signs after the word *Most* in the sentence *Most large pumps work*:



10.3.2 Lexical inference rules

Lexical inference rules add lexical signs.

(Quantified noun ‘everything’ highlights how constraints are applied in modifiers and arguments.)

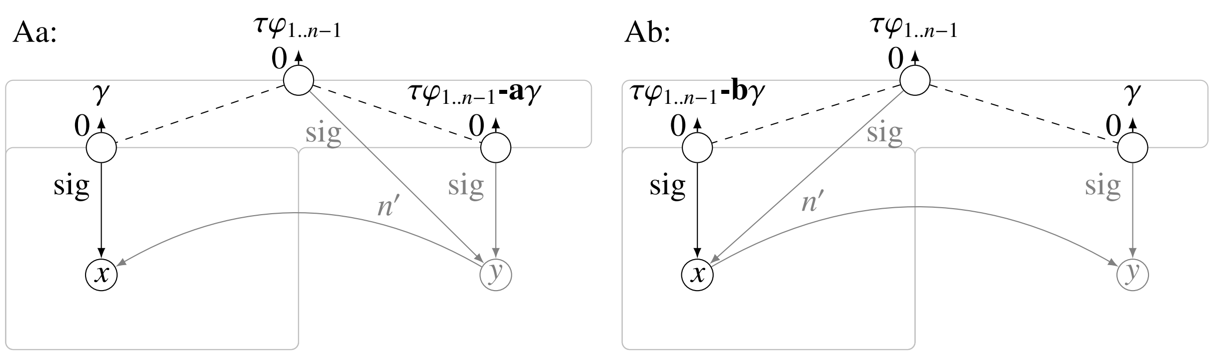


10.3.3 Grammatical inference rules

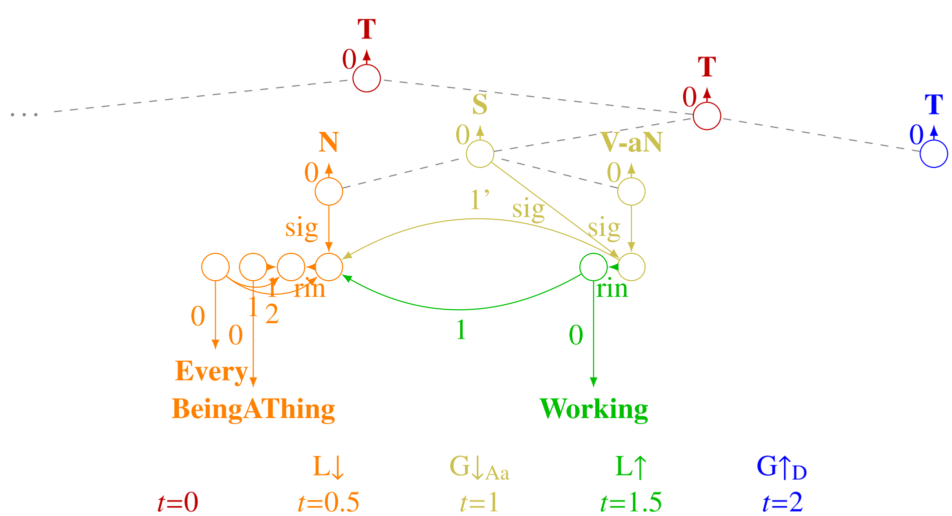
Grammatical inference rules establish associations for syntactic arguments (1′, 2′, . . .).

These form a scaffolding for the participants of predicates, quantifiers, etc.

First we need rules to attach arguments:



These rules attach constraints to the ‘nuclear scopes’ of the quantified noun phrase:



References

de Saussure, F. (1916). *Cours de Linguistique Générale*. Payot.