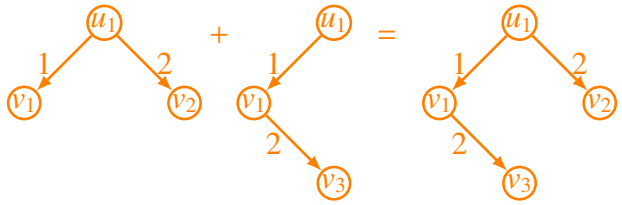
LING5702: Lecture Notes 12
Sentence Processing

We previously saw how meanings could be composed at the computational level, using logic.

This lecture describes an algorithmic-level model of composition in associative memory.

Because associations are just matrices ($\mathbf{M}_1 = \mathbf{v}_1^\top \mathbf{u}_1, \mathbf{M}_2 = \mathbf{v}_2^\top \mathbf{u}_1$), we can unify them by addition:



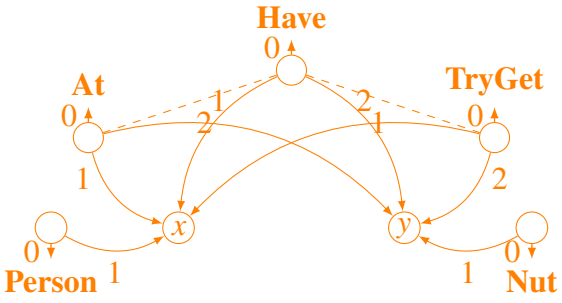
(Some associations may be duplicated, but these can be re-normalized later during generalization.)

We can use this property to build cued association structures during sentence processing!

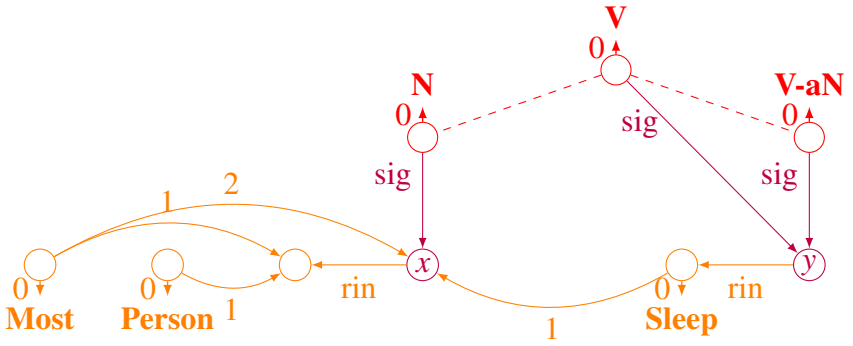
Contents

12.1 Signs [de Saussure, 1916]

Previously we've seen hierarchies of events, which can be modeled using cued associations:



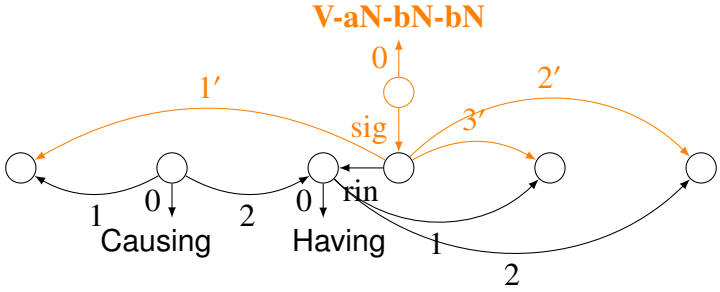
Language may extend this system by using hierarchies of **signs**, which **signify** events:



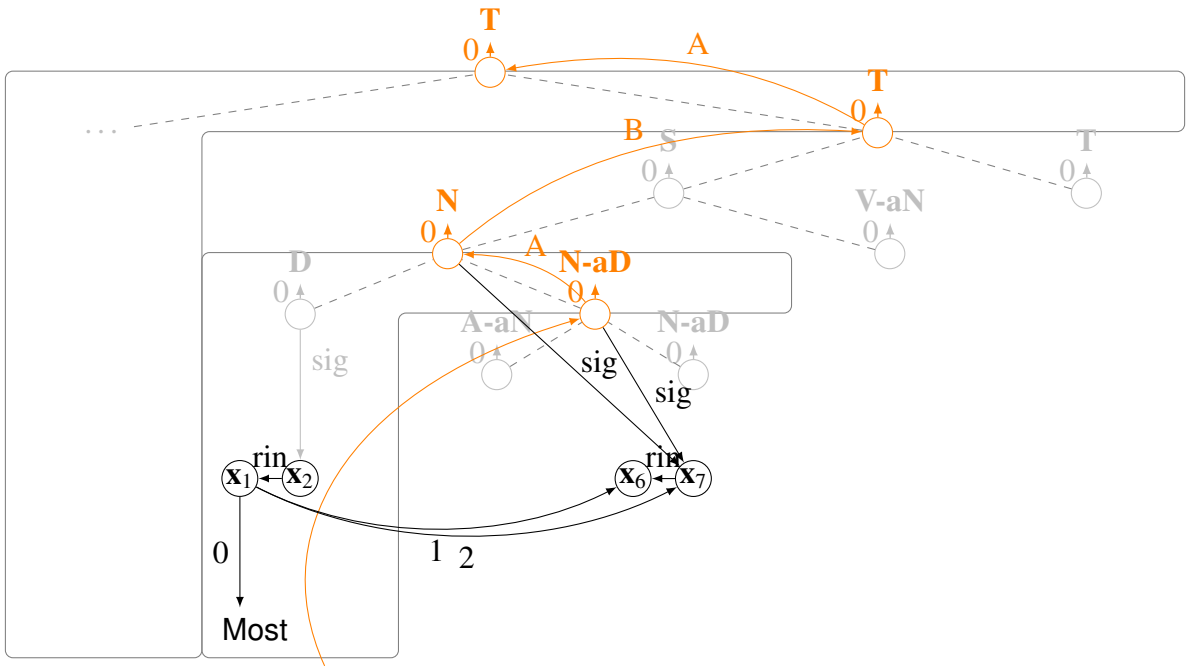
Signs have:

- **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*:

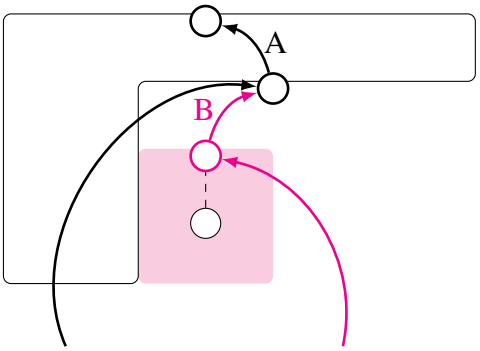


12.2 Processing [Lewis & Vasishth, 2005, Rasmussen & Schuler, 2018]

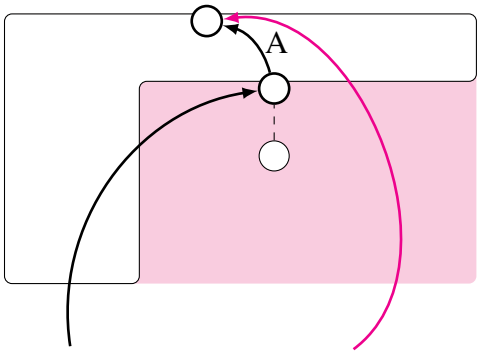
Comprehension proceeds as follows, using modified terminal and nonterminal decisions:

1. a **terminal** decision is made about whether to **match** store elements at the next word, and a **lexical inference rule** is applied (choose a meaning for the next word).

   (a) no terminal (lexical) match:

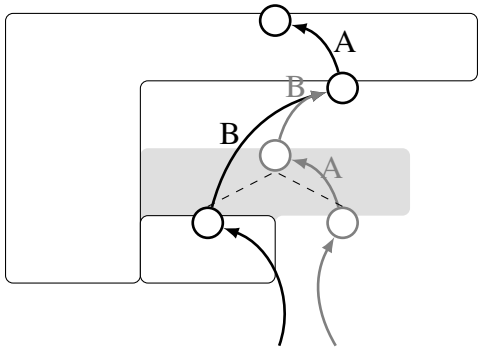
   

   (b) yes terminal (lexical) match:

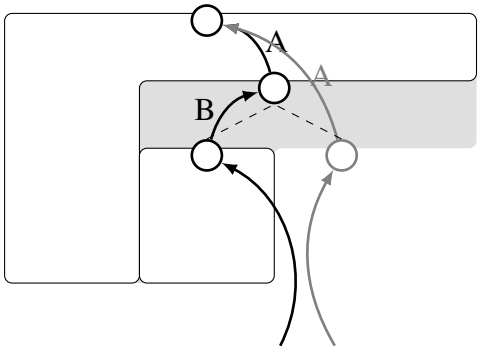
   

2. a **non-terminal** decision is made about whether to **match** store elements at the next rule, a **grammatical inference rule** is applied (choose a rule to compose the next subtree).

(c) no non-terminal (grammatical) match:



(d) yes non-terminal (grammatical) match:

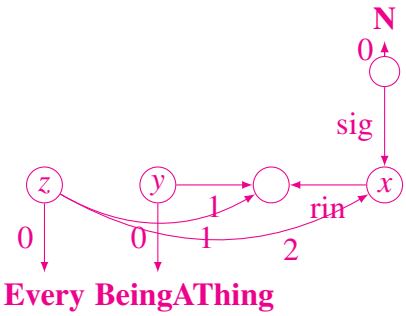


12.3 Lexical inference rules

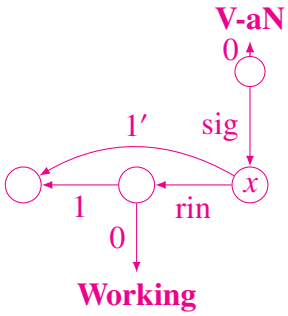
Lexical inference rules add lexical signs.

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

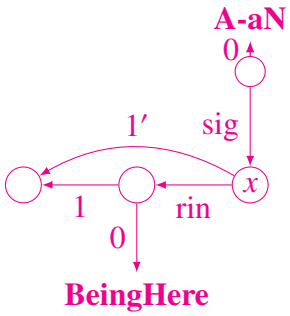
a)



b)



c)



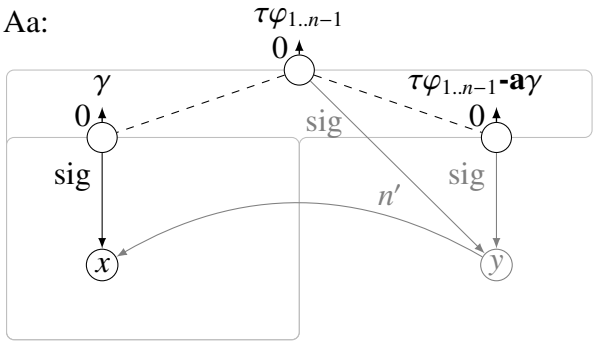
12.4 Grammatical inference rules

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

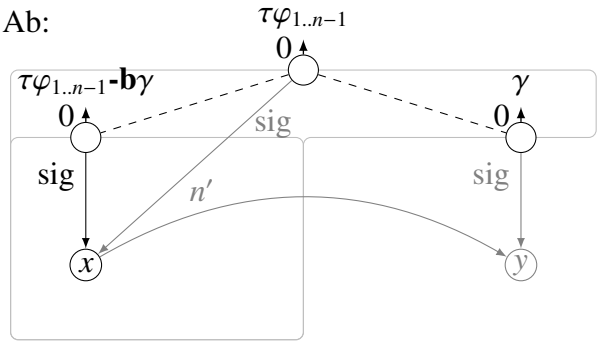
These form a scaffolding for the participants of predicates, quantifiers, etc.

First we need rules to attach arguments:

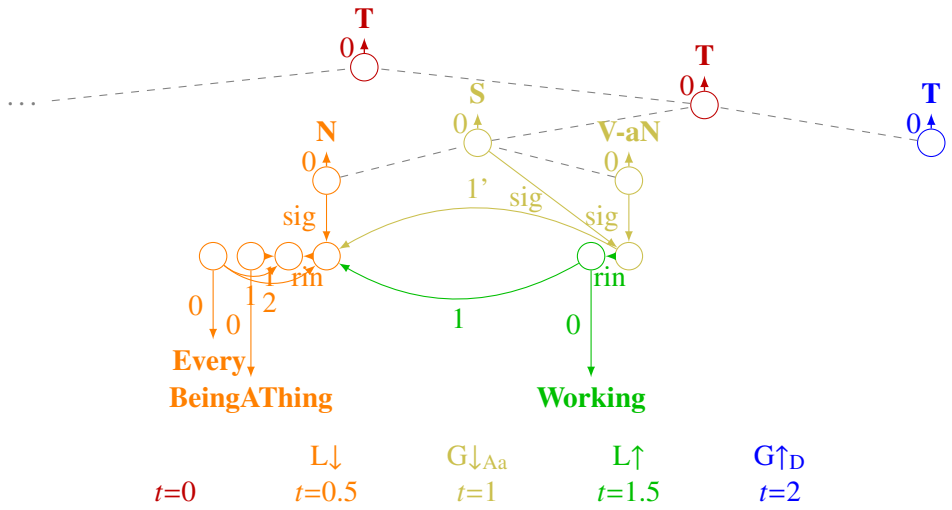
Aa:



Ab:

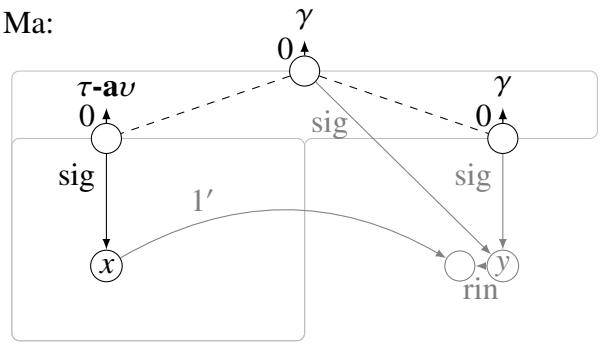


These rules attach constraints to the ‘nuclear scopes’ of the quantified noun phrase:

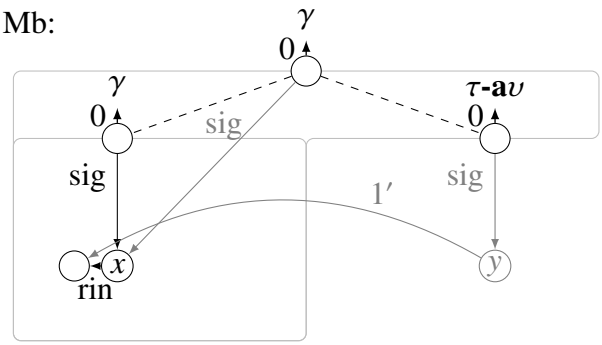


Then we need rules to attach modifiers:

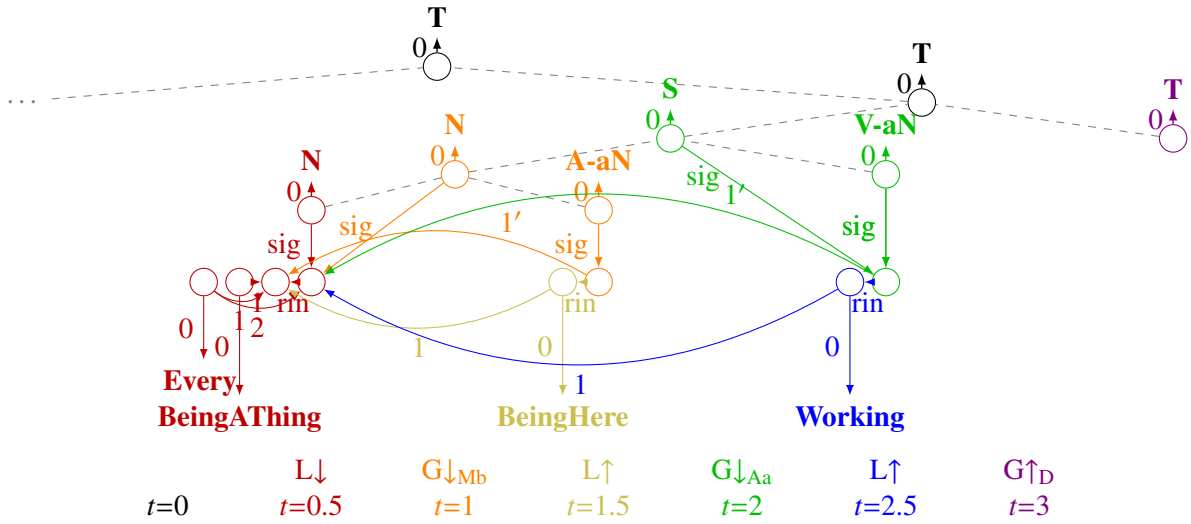
Ma:



Mb:



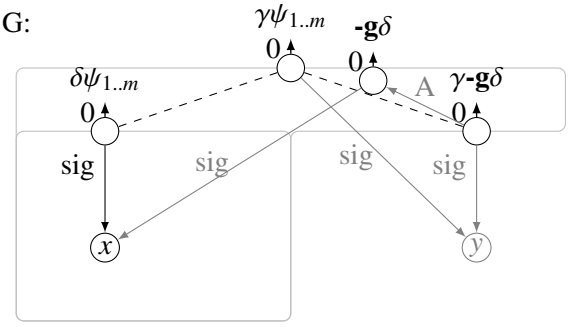
These rules attach constraints to the ‘restrictors’ of quantified noun phrases:



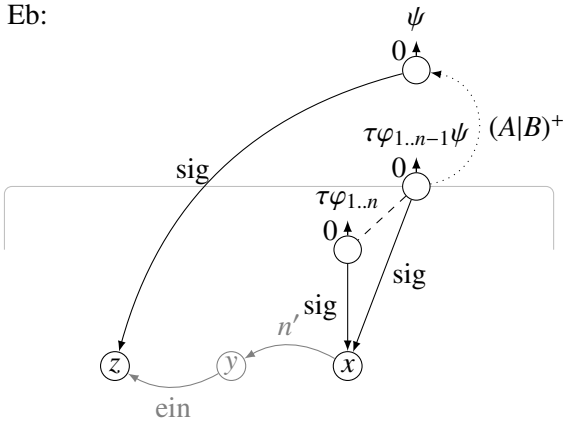
We also need rules to attach fillers to gaps.

The below rules ('G'ap-filler and 'E'xtraction rules) store fillers using apex/base associations:

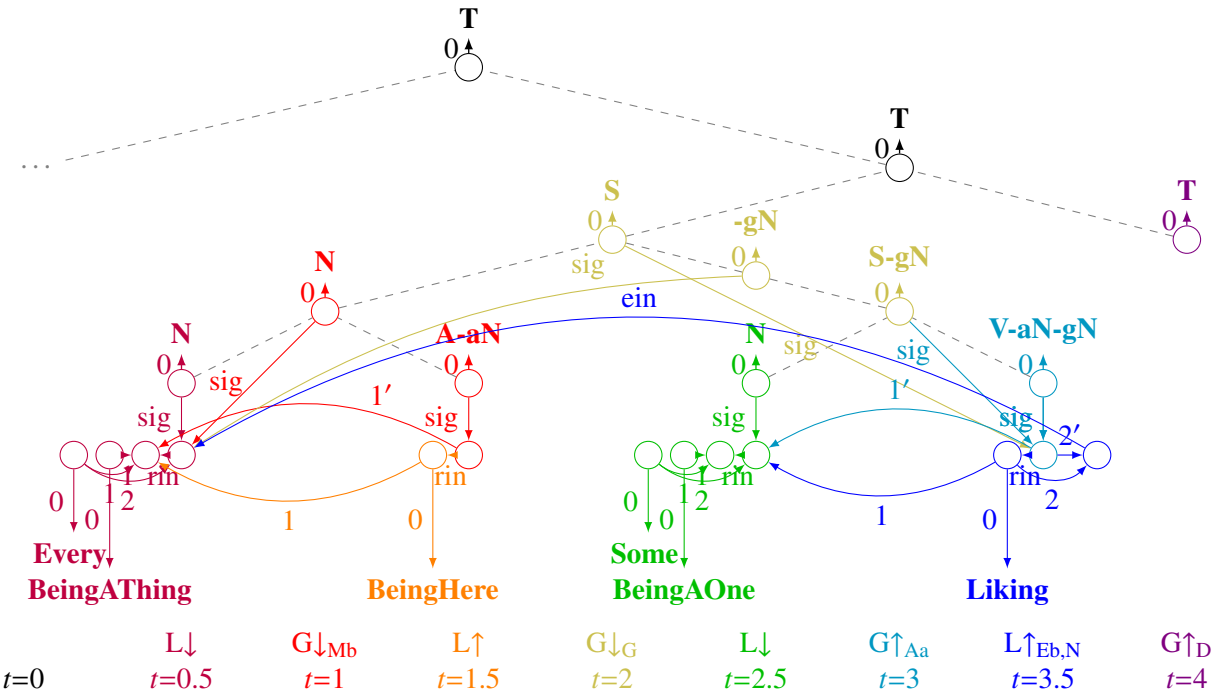
G:



Eb:



Here's an entire derivation of *Everyone here someone likes*:



References

[Lewis & Vasishth, 2005] Lewis, R. L. & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3), 375–419.

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

[Rasmussen & Schuler, 2018] Rasmussen, N. E. & Schuler, W. (2018). Left-corner parsing with distributed associative memory produces surprisal and locality effects. *Cognitive Science*, 42(S4), 1009–1042.