A semantics of subordinate clauses using delayed evaluation

Will Merrill
Linguistics & Computer Science
Yale University, 2019

TULCON 11
March 11th, 2018

Talk outline

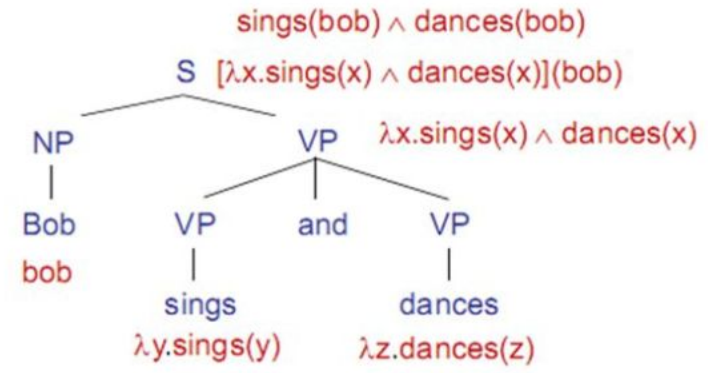
1. Introducing the delayed evaluation formalism
2. Delayed evaluation of subordinate clauses
3. Defining *only* in terms of delayed evaluation

I. The delayed evaluation formalism

Compositional semantics

Meaning of a sentence comes combines meanings of words according to syntactic structure

1. Evaluate meaning of word by looking it up
2. To evaluate meaning of constituent:
   a. Evaluate meanings of children
   b. **Compose** (combine) sub-meanings to get full constituent's meaning
3. Apply recursively to get logical form for the full sentence



Delayed evaluation

Non-compositional rule for computing a constituent’s meaning

At some constituents, we allow the following:

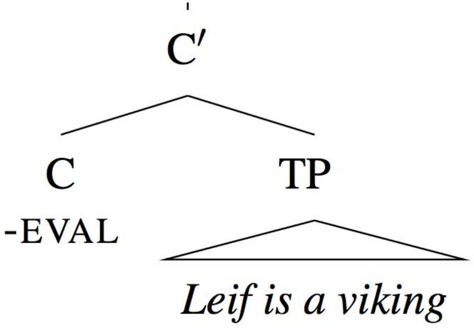
- Don’t evaluate the meaning of the child
- Instead, pass up the unevaluated child node itself
- The child’s meaning can be evaluated later (in different scope/world context)

This will be made more concrete in a second!

Step 1: delaying evaluation

Definition:

If a node α has children $\{\beta, \gamma\}$, and β has the feature -EVAL, then $[[\alpha]]^w = \gamma$.



$$[\![\mathrm{C}']\!]^{w,g} = \mathrm{TP}.$$

- Meaning of higher node is the lower node itself (not its meaning)
- Consequence: Treat syntactic subtrees as semantic objects!
    - Syntactic subtrees get their own semantic type in the lambda calculus

Step 2: evaluating the delayed nodes

- Meanings of words higher in the tree should expect node arguments and evaluate them
- For example, the definition

$$[\![believe]\!]^{w,g} = \lambda p.\lambda x.\forall w'\big(w' \in \mathbb{B}(x) \rightarrow p(w')\big) \in w.$$

p is a lambda predicate that takes a world and returns a truth value

(Fintel & Heim, 2011)

can be re-written

$$[\![believe]\!]^{w,g} = \lambda \alpha.\lambda x.\forall w'(w' \in \mathbb{B}(x) \rightarrow [\![\alpha]\!]^{w',g}) \in w.$$

α is an unevaluated node

II. Delayed evaluation of subordinate clauses

Goal

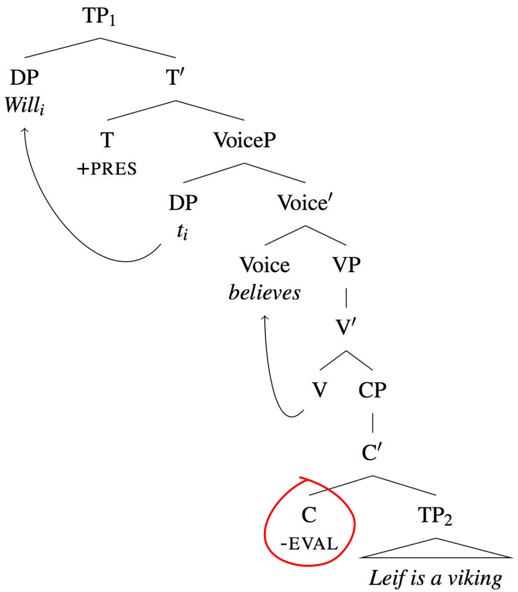
Show how three different kinds of subordinate clauses can be analyzed as special cases of the same general pattern of **delayed evaluation**:

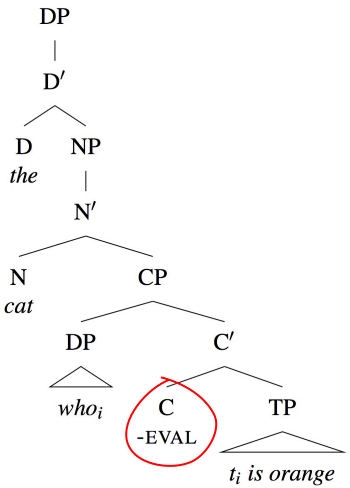
1. Propositional attitude predicates (e.g. *John believes Mary will win*)
2. Relative clauses (e.g. *the cat that is yellow*)
3. Control predicates (e.g. *John wants to win*)

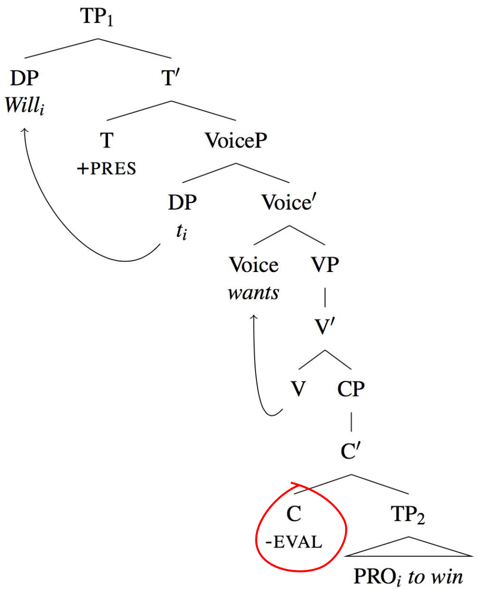
This analysis will eliminate the need for extra rules like predicate abstraction that are otherwise necessary for relative clause semantics

Delayed evaluation in subordinate clauses

- Proposal: Put -EVAL in C to extract the lower clause without evaluating
- We get the right meaning for each construction via standard semantic operations!







Propositional attitude predicates

Evaluate the sense-abstracted node α:

- **In alternative worlds**
- With normal variable scope

Example:

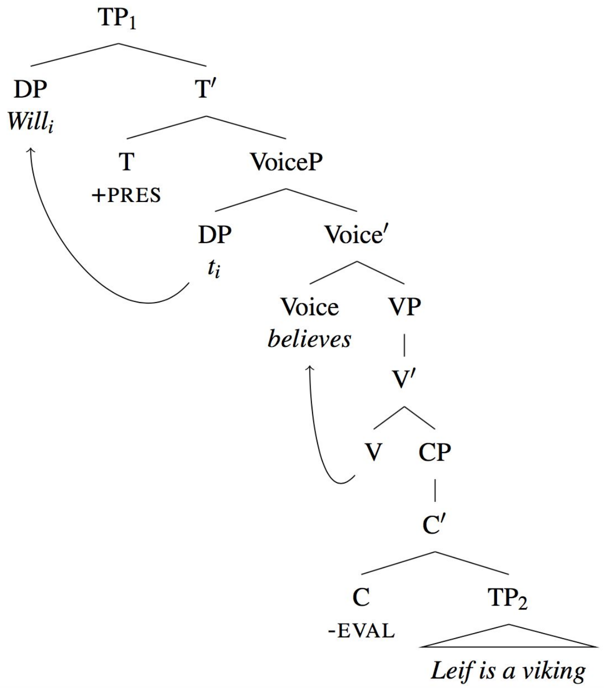
$$[\![believe]\!]^{w,g} = \lambda\alpha.\lambda x.\forall w'(w' \in \mathbb{B}(x) \rightarrow [\![\alpha]\!]^{w',g}) \in w$$

Example: propositional attitude predicates

$$[\![\mathrm{C}']\!]^{w,g} = \mathrm{TP}_2$$

$$[\![\mathrm{V}']\!]^{w,g} = [\![\mathrm{V}]\!]^{w,g}([\![\mathrm{CP}]\!]^{w,g}) = [\![\textit{believes}]\!]^{w,g}([\![\mathrm{C}']\!]^{w,g})$$
$$= \big(\lambda\alpha.\lambda x.\forall w'(w' \in \mathbb{B}(x) \to [\![\alpha]\!]^{w',g} \in w)\big)(\mathrm{TP}_2)$$
$$= \lambda x.\forall w'(w' \in \mathbb{B}(x) \to [\![\mathrm{TP}_2]\!]^{w',g}) \in w$$

$$[\![\mathrm{V}']\!]^{w,g} = \lambda x.\forall w'(w' \in \mathbb{B}(x) \to \mathrm{viking}(\mathrm{Leif}) \in w') \in w$$



Relative clauses

Evaluate the sense-abstracted node α:

- In the extensional world
- **With alternative variable scope**

$$[\![who_i]\!]^{w,g} = \lambda\alpha.\lambda x.[\![\alpha]\!]^{w,(i,x)||g}$$

- *(i,x)||g* is the manipulated variable scope for evaluating the subordinate clause
- By this notation, I mean:
  - Map coindex *i* to *x*
  - Map any other index *j* to *g(j)*

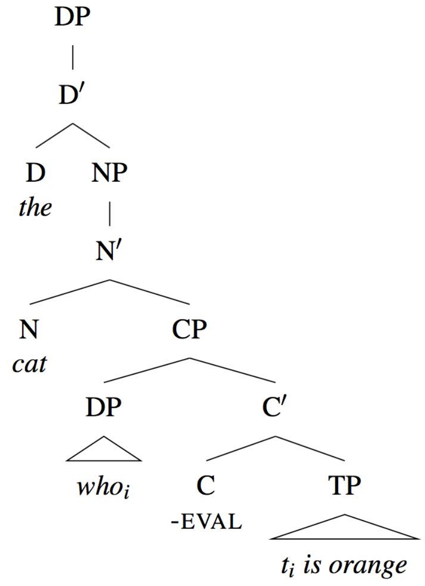
Example: relative clauses

$$[\![\mathrm{CP}]\!]^{w,g} = [\![\mathrm{DP}]\!]^{w,g}([\![\mathrm{C}']\!]^{w,g}) = \left(\lambda\alpha.\lambda x.[\![\alpha]\!]^{w,(i,x)||g}\right)(\mathrm{TP}) = \lambda x.[\![\mathrm{TP}]\!]^{w,(i,x)||g}$$

$$[\![\mathrm{TP}]\!]^{w,g} = [\![t_i\ is\ orange]\!]^{w,g} = \mathrm{orange}(g(i)) \in w$$

$$[\![\mathrm{CP}]\!]^{w,g} = \lambda x.\mathrm{orange}(x) \in w$$

- Conventionally handled with *predicate abstraction* (Heim & Kratzer, 2011)
  - Idiosyncratic non-compositional rule
- Delayed evaluation lets us do predicate abstraction compositionally!



Control predicates

Evaluate the sense-abstracted node α:

- **In alternative worlds**
- **With alternative variable scope**

Example:

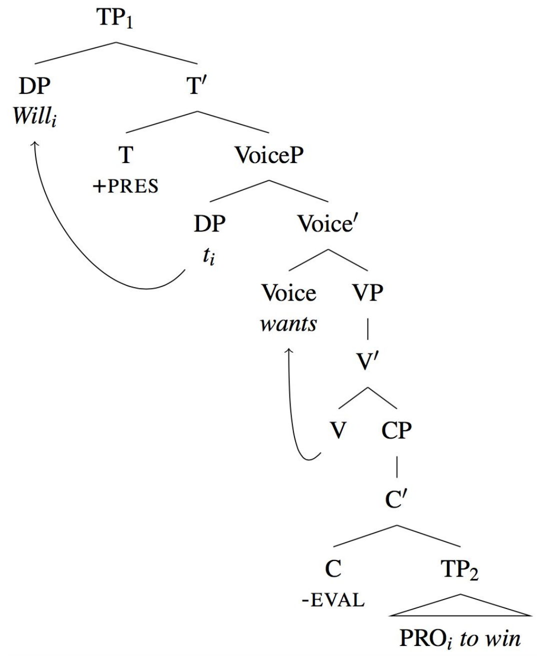
$$[\![want]\!]^{w,g} = \lambda\alpha.\lambda x.\forall w'(w' \in \mathbb{W}(x) \to [\![\alpha]\!]^{w',(i,x)||g}) \in w$$

Example: control predicates

$$[\![\mathrm{V}']\!]^{w,g} = [\![V]\!]^{w,g}([\![CP]\!]^{w,g}) = \big(\lambda\alpha.\lambda x.\forall w'(w' \in \mathbb{W}(x) \to [\![\alpha]\!]^{w',(i,x)||g}) \in w\big)(\mathrm{TP}_2)$$

$$= \lambda x.\forall w'(w' \in \mathbb{W}(x) \to [\![\mathrm{TP}_2]\!]^{w',(i,x)||g}) \in w$$

$$[\![\mathrm{V}']\!]^{w,g} = \lambda x.\forall w'(w' \in \mathbb{W}(x) \to \mathrm{win}(x) \in w') \in w$$



- Empirically, control predicates have null C in deep structure (Bhatt, 2001)
  - Why can't they be small clause TPs?
  - Delayed evaluation provides an answer: need a place for +SENSE to sit!

III. Defining *only* in terms of delayed evaluation

Using delayed evaluation to define *only*

- Can also be used to give a syncategorematic denotation for the word *only*
  - Syncategorematic — meaning combines by standard composition; not irregular rules
- Will need to generalize our definition of delayed evaluation a bit

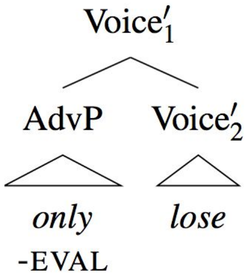
Generalized delayed evaluation (formal definition)

Definition:

If a node α has children {β, ɣ}, and β has the feature -EVAL, then

$[[\alpha]]^w = [[\beta]]^w(\gamma)$.

- Old definition is the special case where $[[\beta]]^w$ is the identity function



$$[\![\text{Voice}'_1]\!]^{w,g} = [\![\text{AdvP}]\!]^{w,g}(\text{Voice}'_2) = [\![\textit{only}]\!]^{w,g}(\text{Voice}'_2)$$

Redefining *only*

- State-of-the-art semantics for *only* gives a “translation rule”:

$$[\![\textit{only}\ \mathrm{VP}]\!]^{w,g} = \lambda x.\forall p\big((p(x) \wedge \mathbb{C}(p)) \rightarrow p = [\![\mathrm{VP}']\!]^{w,g}\big).$$

(Kotek, 2014)

- Delayed evaluation lets us express this as a fully compositional denotation:

$$[\![\textit{only}]\!]^{w,g} = \lambda \alpha.\lambda x.\forall p\big((p(x) \wedge \mathbb{C}_{[\![\alpha]\!]^{w,g}}(p)) \rightarrow p = [\![\alpha]\!]^{w,g}\big).$$

Conclusion

1. Delayed evaluation can be used to model semantics of subordinate clauses
    a. Attitude predicates, relative clauses, and control predicates correspond to the three possible “use cases” for unevaluated nodes
2. Unevaluated nodes are similar to structured intensions
3. Can express a compositional denotation for *only* in terms of delayed evaluation

Acknowledgments

Thanks (!) to:

- Yale Semantics Reading Group
- TULCON 11 organizers