A semantics of subordinate clauses using delayed evaluation

1 Abstract

We suggest a general semantic treatment of several subordinate clause types grounded in their syntactic similarity. Crucial to this framework is the choice to view syntactic trees as evaluatable objects that can be passed as meanings through the lambda calculus. Thus, the meaning of a subordinate clause is a pointer to a syntactic node whose denotation can be evaluated further along in the process of semantic composition. This analysis is sufficiently general to unify the semantic treatment of propositional attitude predicates, relative clauses, and control predicates without the need for exceptional rules like predicate abstraction. It can also be used to express a syncategorematic denotation for *only*.

2 Introduction

A simple definition of an intension is a function that takes a world argument, or, in other words, $\lambda w.f(w)$ for some f.[5], [7] By this definition, a proposition is a special case of an intension when $f(w)$ returns a truth value. Each proposition can be thought of as a characteristic function that corresponds to a unique set of possible worlds.

This property of propositions turns out to be problematic for representing the semantics of propositional attitude predicates like the verb *believe*.[3] Historically, one way of accounting for this has been the notion of "structured propositions".[8] Such theories define propositions as hierarchically structured objects that determine a set of worlds. This solves the problem of propositional attitude predicates because two propositions with different structure are unequal even when the set of worlds they determine can be the same. An equivalent way of describing this is that propositional attitude predicates are sensitive not just to the λw-intension of their clause argument, but also the λw-intensions of constituents within the clause.[3]

The core idea of this paper also adds structure to intensions, but we go about it with different reasons and methods. We define intensions as pointers to syntactic subtrees. This means that the structure of an utterance's intension is just the syntactic structure of the utterance. To evaluate an intension α in world w, we can take $[\![\alpha]\!]^w$. We show that this formulation of intensions provides a theoretically cohesive treatment of the semantics of several different subordinate clause types.

In order to drive this theory of intensionality, we introduce a feature -EVAL that can sit at phrasal heads (most commonly, C). The effect of -EVAL is to intensionalize the semantics of its sibling node, which we define the following compositional rule to encode:

Definition 2.1 (General delayed evaluation)**.** *If a node* α *has children* $\{\beta, \gamma\}$ *and* β *has the feature* -EVAL*, then*

$$[\![\alpha]\!] = [\![\beta]\!](\gamma).$$

Corollary 2.1 (Special delayed evaluation)**.** *If a node* α *has children* $\{\beta, \gamma\}$*,* β *has the feature* -EVAL*, and* $[\![\beta]\!] = \mathrm{Id}$*, then*

$$[\![\alpha]\!] = \gamma.$$

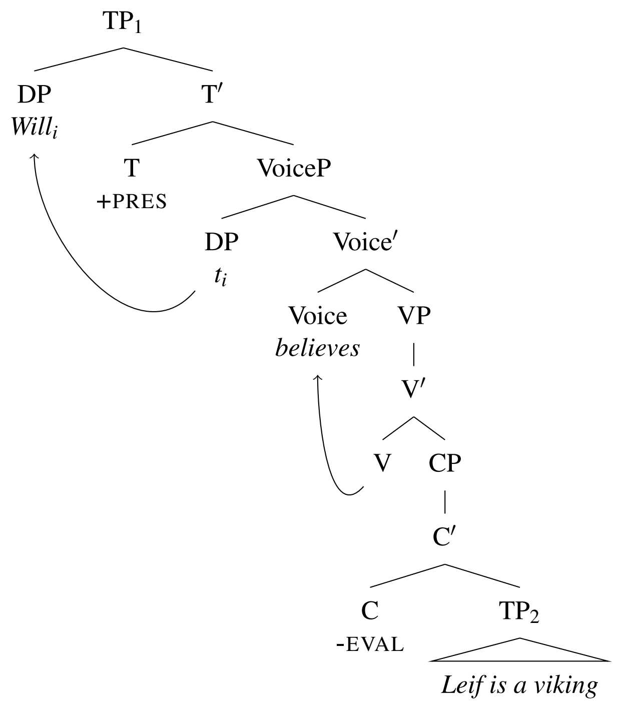
This rule takes precedence over function application, which we assume is the default compositional rule for binary-branching nodes. For comparison, function application computes the denotation of each child and then applies the resulting lambda expressions in whatever order is possible.[5] Sense abstraction resembles this, but we only compute the denotation of the child marked with -EVAL and then apply a pointer to the other child as its argument. Since β must have been marked with -EVAL to trigger predicate abstraction, it is assumed that its denotation is defined to handle a syntactic tree as an argument.

In most cases, the base semantics of the node β are the identity function, so $[\![\alpha]\!] = \mathrm{Id}(\gamma) = \gamma$. Thus, the effect of -EVAL is to return an intensional version of its sibling node.

3 Propositional attitude predicates

As has been mentioned, propositional attitude predicates were an important case in the development of structured intensions.[3], [8] The motivation for moving away from unstructured sets of worlds goes as follows: the sentences *John rolls a* 4 and *John rolls a* $2+2$ are true in exactly the same set of worlds, but it is possible that some agent believes one and not the other.

However, recent comprehensive work in intensional semantics has avoided using structured intensions for such cases.[6] Instead, one can remedy the problem with vanilla possible world semantics by allowing inconsistent and incomplete worlds into the set of worlds quantified over.[1] In the following example derivation, we choose to adapt this treatment into the terms of sense abstraction for the sake of familiarity. In principle, one might be able to use the structure of sense-abstracted nodes in a semantics of belief, but such a goal is outside the scope of our current argument.



By the rule for delayed evaluation, the denotation of C′ is given by

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

This value passes up the tree to CP. At this point, we need to adapt the standard denotation of *believe* given in (3.1) to the framework of delayed evaluation by replacing the λw-predicate p with a pointer to a node.

Definition 3.1 (Conventional denotation of *believe*)**.** [6]

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

Definition 3.2 (Our denotation of *believe*)**.** [6]

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

We assume that worlds are sets of true propositions, so stating that a proposition is in a world means that the proposition is true in that world. The expression $\mathbb{B}(x)$ returns a set of worlds that x believes are possible, and its value is different depending on the world. Placing an expression containing $\mathbb{B}(x)$ inside the world w means that the expression references the value of $\mathbb{B}(x)$ in w.

Plugging in the denotation of *believe*, we get that

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

We now evaluate the denotation of TP_2 and use it to simplify this expression:

$$\begin{aligned}
[\![\mathrm{TP}_2]\!]^{w,g} = [\![\textit{Leif is a viking}]\!]^{w,g} = \mathrm{viking}(\mathrm{Leif}) \in w. \\
[\![\mathrm{V}']\!]^{w,g} = \lambda x.\forall w'(w' \in \mathbb{B}(x) \to \mathrm{viking}(\mathrm{Leif}) \in w') \in w.
\end{aligned}$$

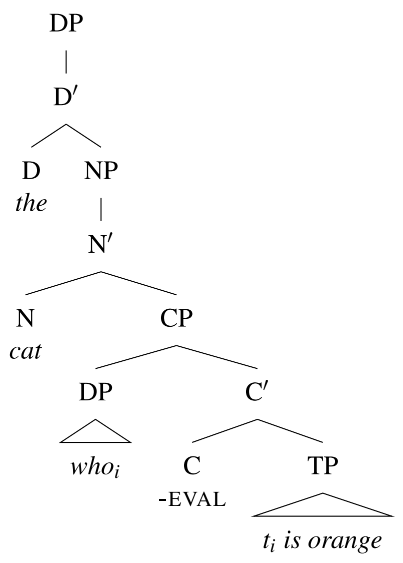
In order to get the semantics of the sentence, we substitute Will for x. For simplicity, any temporal logic semantics that would be introduced by T have been omitted. The end result is is the logical form

$$[\![\mathrm{TP}_1]\!]^{w,g} = \big(\forall w'(w' \in \mathbb{B}(\mathrm{Will}) \to \mathrm{viking}(\mathrm{Leif}) \in w')\big) \in w.$$

The case of propositional attitude predicates illustrates how delayed evaluation of syntactic nodes can fulfil the same function as intensions of the form $\lambda w.f(w)$. We now turn to several related cases where sense abstraction seems to provide a general solution to semantic phenomena that is not achievable with λw-intensions.

4 Relative clauses

Delayed evaluation provides an elegant solution to computing the semantics of relative clauses. We arrive at the correct denotation for a relative clause by assigning -EVAL to C just as we did for belief clauses. This reflects a fundamental structural symmetry between the two constructions. Furthermore, it eliminates the need for rules like predicate abstraction that, conventionally speaking, have been necessary to derive relative clause semantics.



As in the previous example, we get that

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

Now, let $(i,x)||g$ be the function mapping i to x and j to $g(j)$ for $j \neq i$. We give a syncategorematic definition for *who* that will compose with the node pointer and yield the desired semantics:

Definition 4.1 (Denotation of *who*$_i$)**.**

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

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

We can now evaluate $[\![\mathrm{TP}]\!]^{w,(i,x)||g}$ and simplify our expression:

$$[\![\mathrm{TP}]\!]^{w,g} = [\![t_i \textit{ is orange}]\!]^{w,g} = \mathrm{orange}(g(i)) \in w.$$
$$\therefore [\![\mathrm{TP}]\!]^{w,(i,x)||g} = \mathrm{orange}(x) \in w.$$
$$[\![\mathrm{CP}]\!]^{w,g} = \lambda x.\mathrm{orange}(x) \in w.$$

This is the standard denotation that would be assigned to this clause by a rule like predicate abstraction.[6] We can then go on to combine this predicate with the denotation of *cat* via predicate modification.[6] Finally, we get that the denotation of the full sentence is given by

$$[\![\mathrm{DP}]\!]^{w,g} = \iota x.(\mathrm{cat}(x) \in w \wedge \mathrm{orange}(x) \in w).$$

Without considering relative clauses, one might devise a semantics for -EVAL that abstracts a λw over its sibling node and thus resembles conventional formulations of intensions.[5], [7] This contrasts with our treatment, in which the λw-intension is replaced by a pointer to an executable syntactic node. Therefore, if we assume that -EVAL is present at C for relative clauses, then they provide a testable case that we can use to distinguish between these two hypotheses. As we will see, only the latter produces the right semantics for relative clauses.

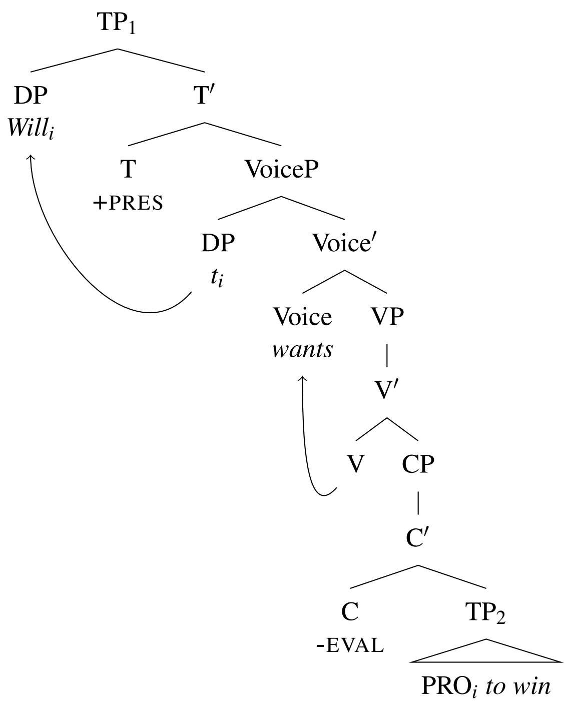
The main difference between the semantics of relative clauses and of belief clauses is that the former is abstracting over variable assignments while the latter is abstracting over worlds. Lambda-theoretic intensionality can handle world abstraction adequately since intensions are defined to be functions that take world arguments. On the other hand, such functions cannot represent abstraction over variable assignments. This is why additional assumptions like predicate abstraction have traditionally been needed to handle relative clause semantics.

Delaying evaluation provides a general way to handle both belief clauses and relative clauses. We delay the evaluation of the relative clause until we have the correct variable scope just like we delay the evaluation of a belief clause until we have the correct world. The two subordinate clauses actually have the same structure (-EVAL at C). What distinguishes the semantics of these two clause types is what context higher nodes construct for evaluating the lower clause. With belief clauses, the matrix predicate quantifies over alternative worlds, whereas, with a relative clause, the operator in the specifier of CP constructs an alternative variable scope.

5 Control predicates

Placing -EVAL at the C head is a general way of handling the semantics of many different types of subordinate clauses. This same treatment also works for control predicates, which combine the modal characteristics of verbs like *believe* with the coreference of relative clause constructions. Within our formalism, this corresponds to evaluating the subordinate clause both with modified variable scope and in an alternative world. Thus, these predicates constitute a third case exhausting the range of uses that we would expect for evaluating clauses with -EVAL.

It has been demonstrated in the syntactic literature that English control predicates never take TPs.[2] Instead, their complement clause is a full CP with a null C head. Our formalism explains this phenomenon because it predicts that we always need so that -EVAL has somewhere to sit. With this in mind, consider the example sentence *Will wants to win*:



The rule for delayed evaluation gives us that

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

We now give a denotation for *want* that hybridizes the denotations of *believe* in (3.2) and *who* in (4.1).

Definition 5.1 (Denotation of *want*)**.**

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

Applying this yields that

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

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

Observe that

$$[\![\mathrm{TP}_2]\!]^{w,g} = \mathrm{win}(g(i)) \in w.$$
$$\therefore [\![\mathrm{TP}_2]\!]^{w',(i,x)||g} = \mathrm{win}(x) \in w'.$$

.

We use the denotation for TP to get the semantics for the sentence.

$$[\![\mathrm{V}']\!]^{w,g} = \lambda x.\forall w'(w' \in \mathbb{W}(x) \to \mathrm{win}(x) \in w') \in w.$$
$$\therefore [\![\mathrm{TP}_1]\!]^{w,g} = \forall w'(w' \in \mathbb{W}(\mathrm{Will}) \to \mathrm{win}(\mathrm{Will}) \in w') \in w.$$

6 The alternative quantifier *only*

Interestingly, sense abstraction is also useful when trying to write explicit semantics for *only*: a problem that has received considerable literature attention. One of the first major insights was Laurence Horn's treatment, which pioneered using mixed presuppositional and assertional semantics.[5] Later, Mats Rooth refined the formulation of the assertional component in terms of alternative semantics.[10], [11] Rooth's semantics are idiosyncratic in that they do not provide a lexical denotation for *only* as we would like to have in a fully compositional system. In other words, they define the meaning of the internal node spanning *only* and its argument, but not the meaning of *only* itself. Take, for example, the following "translation rule" adapted from Hadas Kotek's reformulation of Rooth's semantics:[9]

Definition 6.1 (Translation rule for *only*)**.**

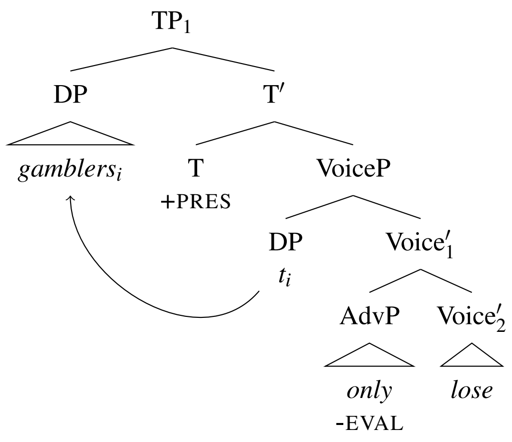
$$[\![\textit{only}\ \mathrm{VP}]\!]^{w,g} = \lambda x.\forall p\big((p(x) \wedge \mathbb{C}(p)) \to p = [\![\mathrm{VP}']\!]^{w,g}\big).$$

Here, we assume that $\mathbb{C}$ is the characteristic function over a contextually determined alternative set for the verbal predicate. Thus, it functions as a "domain selector" for which actions are appropriate alternatives to the predicate within the discourse context. Sense abstraction enables us to convert this translation rule to a standard lexical denotation:

Definition 6.2 (Denotation of *only*)**.**

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

This definition gives us the correct denotation for *only* when we also give the structural position holding *only* the -EVAL feature. Consider the following:



This is a case of delayed evaluation where the denotation of the structural position with -EVAL is not the identity function. By the general definition of delayed evaluation in (2.1), we get:

$$
\begin{gathered}
[\![\text{Voice}'_1]\!]^{w,g} = [\![\text{AdvP}]\!]^{w,g}(\text{Voice}'_2) = [\![\textit{only}]\!]^{w,g}(\text{Voice}'_2) \\
= \big(\lambda\alpha.\lambda x.\forall p\big((p(x) \wedge \mathbb{C}_{[\![\alpha]\!]^{w,g}}(p)) \to p = [\![\alpha]\!]^{w,g}\big)\big)(\text{Voice}'_2) \\
= \lambda x.\forall p\big((p(x) \wedge \mathbb{C}_{[\![\text{Voice}'_2]\!]^{w,g}}(p)) \to p = [\![\text{Voice}'_2]\!]^{w,g}\big) \\
= \lambda x.\forall p\big((p(x) \wedge \mathbb{C}_{\lambda y.\text{lose}(y)\in w}(p)) \to p = \lambda y.\text{lose}(y) \in w\big). \\
\therefore\ [\![\text{TP}]\!]^{w,g} = \forall p\big((p(\text{gamblers}) \wedge \mathbb{C}_{\lambda y.\text{lose}(y)\in w}(p)) \to p = \lambda y.\text{lose}(y) \in w\big).
\end{gathered}
$$

Because our delayed evaluation rule is type-generic, the definition given in (6.2) should be able to account for the semantics of *only* no matter what type of phrase it adjoins into. For example, in addition to this example with a VP, it would also produce the desired semantics for a DP like *the only king*.

7 Conclusion

Delayed evaluation offers a general solution to the semantics of several different constructions involving subordinate clauses. In each case, giving C the -EVAL feature yields the correct denotation for the construction. The three clause types discussed represent the three different ways that an unevaluated constituent can be used. Predicates like *believe* evaluate the node in alternate worlds. In the case of a relative clause, the node is evaluated with a modified variable assignment function to produce the correct coreferential semantics. Finally, in the case of control predicates, both an alternative world and a modified variable assignment scope are passed while evaluating. Thus, delayed evaluation is a theoretically coherent way of explaining the semantics of several different relative clause types that have previously been hard to connect.

In addition, delayed evaluation offers an elegant way of writing a syncategorematic denotation for *only*. While this is structurally quite different from the cases with subordinate clauses, the same -EVAL feature can be used to explain the semantics of both types of constructions.

References

[1] Berto, Francesco. *Impossible worlds*, The Stanford Encyclopedia of Philosophy (Winter 2013 Edition). Edward N. Zalta (ed.). <https://plato.stanford.edu/archives/win2013/entries/impossible-worlds/>.

[2] Bhatt, R. (2001). Lecture notes in syntax, Ms. 5.

[3] Cresswell, M.J. (1985). *Structured meanings*. Cambridge, MA: MIT Press.

[4] Frege, G. (1948). *Sense and reference*. The Philosophical Review, 57(3), 209-230.

[5] Fintel, K. V., & Heim, I. (2011). Lecture notes in intensional semantics, Ms.

[6] Heim, I., & Kratzer, A. (1998). *Semantics in generative grammar*, 13(1). Oxford: Blackwell.

[7] Higginbotham, James. (2003). *Conditionals and compositionality*. Philosophical Perspectives (Vol. 17). 181-194.

[8] King, Jeffrey C. *Structured propositions*. The Stanford Encyclopedia of Philosophy (Fall 2017 Edition). Edward N. Zalta (ed.). <https://plato.stanford.edu/archives/fall2017/entries/propositions-structured/>.

[9] Kotek, H. (2014). *Unifying focus*, Ms., 1.

[10] Rooth, M. (1985). *Association with focus*.

[11] Rooth, M. (1992). *A theory of focus interpretation*. Natural language semantics, 1(1), 75-116.

[12] Horn, L. (1969). *A presuppositional approach to only and even*. Chicago Linguistic Society (5), 98-107.