Sense abstraction: a generalization of intensionality for the semantics of subordinate clauses

1 Abstract

We suggest a general semantic treatment of several subordinate clause types grounded their syntactic similarity. Crucial to this framework is the choice to view syntactic trees as executable objects that can be passed as arguments through the lambda calculus. This departs from the conventional view that intensions are lambda-theoretic expressions of the form $\lambda w.f(w)$.[3], [5] Instead, we treat them as pointers to syntactic nodes whose denotation can be evaluated further along in the process of semantic composition. This analysis is sufficiently general to unify the treatment of the semantics of propositional attitude predicates, relative clauses, and control predicates without the need for exceptional rules like predicate abstraction. It also explains several empirical observations in syntax and offers an elegant way to represent the semantics of "only".

2 Introduction

In order to drive this theory of intensionality, we introduce a feature +SENSE that can sit at phrasal heads (most commonly, C).[1] The effect of +SENSE is to intensionalize the semantics of its sibling node. In order to encode this effect, we define the following compositional rule:

Definition 2.1 (Sense abstraction)**.** *If a node α has children $\{\beta, \gamma\}$ and β has the feature* +SENSE,[2] *then* $[\![\alpha]\!] = [\![\beta]\!](\gamma)$.

This rule takes precedence over function application, which we assume is the default compositional rule for binary-branching nodes. For comparison, function application says the following:[3]

Definition 2.2 (Function application)**.** *If a node α has children $\{\beta, \gamma\}$ and $[\![\beta]\!]$ has $[\![\gamma]\!]$ in its domain, then* $[\![\alpha]\!] = [\![\beta]\!]([\![\gamma]\!])$.

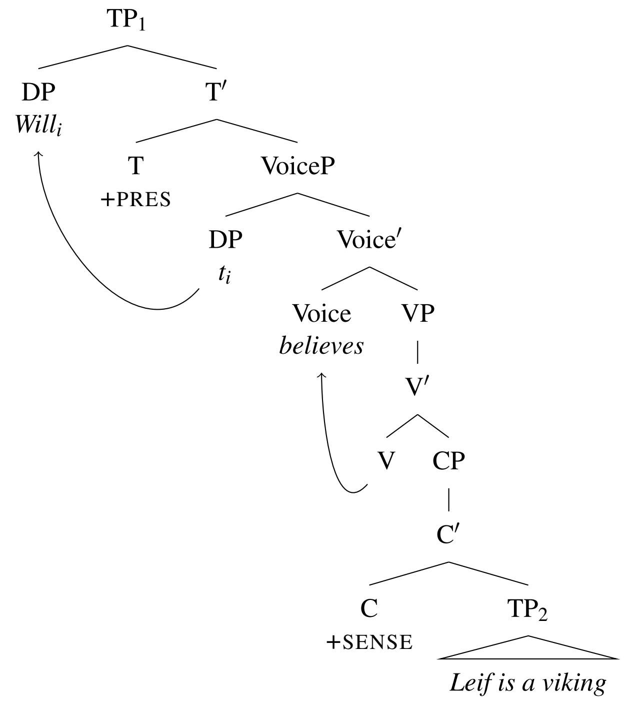
Function application computes the denotation of each child and then applies the resulting lambda expressions in whatever order is possible. Sense abstraction resembles this, but we only compute the denotation of the child marked with +SENSE and then apply a pointer to the other child as its argument. Since β must have been

[1]The term "sense" is taken from Frege, where, in contrast with "reference", it refers to the algorithmic component of meaning.[2]

[2]To emphasize the symmetry with function application, we could also formulate this as "if $[\![\beta]\!]$ has γ in its domain". The definition in terms of +SENSE is presented because it seems more intuitive.

marked with +SENSE to trigger predicate abstraction, it is assumed that its denotation is defined to handle a syntactic tree as an argument.

In simple cases (such as with propositional attitude predicates), the base semantics of the node β are the identity function, so $[\![\alpha]\!] = \mathrm{Id}(\gamma) = \gamma$. Thus, the effect of +SENSE is to return an intensional version of its sibling node. For example, consider the sentence *Will believes Leif is a viking*:



By sense abstraction, we get the denotation of C′ to be a pointer to TP$_2$:

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

This value bubbles up the tree to CP, where it is applied to the denotation of *believe*:

Definition 2.3 (Denotation of *believe*)**.**

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

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 the proposition containing $\mathbb{B}(x)$ inside the world w means that the proposition is referencing the value of $\mathbb{B}(x)$ in w. Plugging the denotation of *believe* into our compositional semantics, we get:

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

We can use standard standard compositional rules to compute the semantics of the embedded clause:

$$[\![\mathrm{TP}_2]\!]^{w,g} = [\![\textit{Leif is a viking}]\!]^{w,g} = \mathrm{viking}(\mathrm{Leif}) \in w$$

Plugging this into the denotation of V$'$ yields:

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

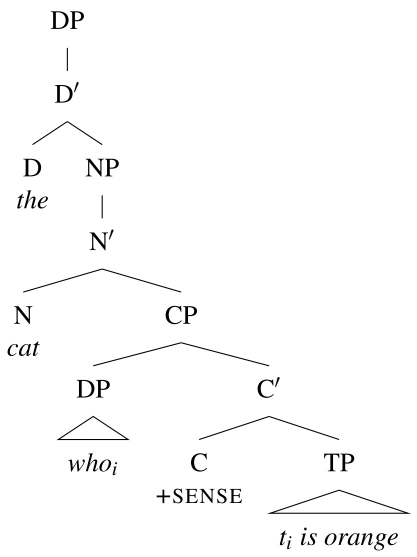
In order to get the semantics of the sentence, we substitute Will for x. For simplicity, any temporal logic semantics that would be introduced by T have been omitted. The end result is the following logical form:

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

This example with a propositional attitude predicate illustrates how sense abstraction can account for intensional statements similarly to conventional intensions of the form $\lambda w.f(w)$. We will 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.

3 Relative clauses

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



As in the previous example, we start be sense abstracting the lower clause:

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

Next, we will define the semantics for *who$_i$*. Let $(i,x)||g$ be the function mapping i to x and j to $g(j)$ for $j \neq i$.

Definition 3.1 (Denotation of *who$_i$*)**.**

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

Applying this definition, we get:

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

Since $[\![\mathrm{TP}]\!]^{w,g} = [\![t_i \textit{ is orange}]\!]^{w,g} = \mathrm{orange}(g(i)) \in w$, we see that $[\![\mathrm{TP}]\!]^{w,(i,x)||g} = \mathrm{orange}(x) \in w$. Therefore, the denotation of CP reduces to:

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

This is the standard denotation that would be assigned to this clause by a rule like predicate abstraction.[4] We can then go on to combine this predicate with the denotation of *cat* via predicate modification.[4] Finally, incorporating the definite article yields the following result for the semantics of the full DP:

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

Without considering relative clauses, one could devise a semantics for +SENSE that abstracts a λw over its sibling node and thus resembles conventional formulations of intensions.[3], [5] This contrasts with our treatment, in which the lambda-theoretic intensional representation is replaced with a pointer to an executable syntactic node. Therefore, if we assume that +SENSE 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, the latter is the only one that produce the right semantics for relative clauses.

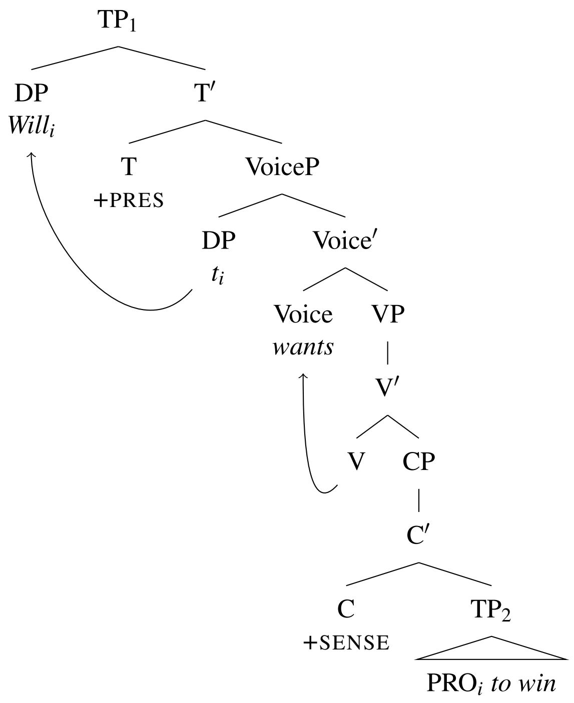
The main difference between the semantics of relative clauses and belief clauses is that the former is abstracting over variable assignments while the latter is abstracting over worlds. Lambda-theoretic intensionality can handle world abstraction adequately since intensions are defined to be functions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

Sense abstraction provides a general way to handle both belief clauses and relative clauses. We are able to 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 (+SENSE at C) in both cases. What distinguishes the semantics of these two clause types is what context higher nodes construct for evaluating the sense-abstracted 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 assignment for the evaluation of the subordinate clause.

4 Control predicates

Placing +SENSE 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 coreferential semantics of relative clauses. In terms of sense abstraction, 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 possible uses that we would expect for sense abstracted clauses.

It has been demonstrated in the syntactic literature that English control predicates never take TPs.[1] Instead, their complement clause is a full CP with a null C head. This observation is explained by sense abstraction, which predicts that we always need a C node with +SENSE in order to derive the correct semantics. With this in mind, consider the example sentence *Will wants to win*:



We start with identical sense abstraction to the previous examples:

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

We will define the denotation of *want* as follows:

Definition 4.1 (Denotation of *want*)**.**

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

Observe that this definition hybridizes the denotations of *believe* in (2.3) and *who* in (3.1). We will proceed by computing the value of V′ compositionally:

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

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

At this point, we see that $[\![\mathrm{TP}_2]\!]^{w,g} = \mathrm{win}(g(i)) \in w$. Therefore, $[\![\mathrm{TP}_2]\!]^{w',(i,x)||g} = \mathrm{win}(x) \in w'$. This leaves us with:

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

By applying the referent of the subject DP to this, we are left with the correct denotation for the matrix clause, *i.e.*:

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

5 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.[3] Later, Mats Rooth gave a more explicit formulation of *only*'s assertional denotation in terms of alternative semantics.[7], [8] While Rooth's semantics are accurate, they 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:[6]

Definition 5.1 (Translation rule for *only*)**.**

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

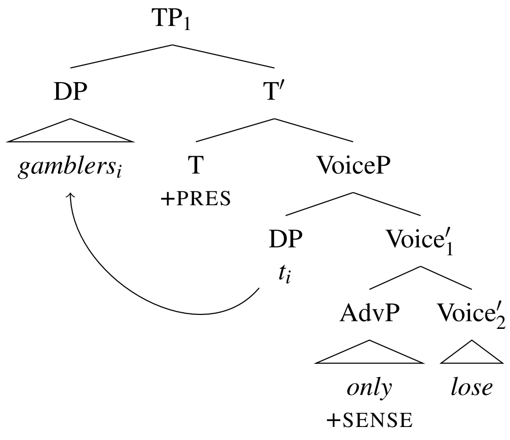
Here, we assume that $\mathbb{C}$ is the characteristic function over the alternative set of the verbal predicate. Thus, it functions as a "domain selector" that determines what alternative semantics are contextually appropriate for a predicate.

Sense abstraction enables us to convert this translation rule to a standard lexical denotation:

Definition 5.2 (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 +SENSE feature. For example, consider the following sentence:



This is a case of sense abstraction where the denotation of the structural position with +SENSE is not the identity function. By the general definition of sense abstraction in (2.1), we get:

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

It follows through normal function application that the assertion of the root node is:

$$[\![\text{TP}]\!]^{w,g} = \forall p\big((p(\text{gamblers}) \wedge \mathbb{C}_{\lambda y.\text{lose}(y)\in w}(p)) \to p = \lambda y.\text{lose}(y) \in w\big)$$

Because sense abstraction is type-generic, the definition given in (5.2) 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 other king*.

6 Conclusion

Sense abstraction offers a general solution to the semantics of several different constructions involving subordinate clauses. In each case, giving C the +SENSE feature yields the correct denotation for the construction. The three clause types discussed represent the three different ways that the node pointer returned by sense abstraction can be used. Predicates like *believe* evaluate the node over possible alternate worlds. In the case of a relative clause, the sense-abstracted node is evaluated with a modified variable assignment function that produces the correct coreferential semantics. Finally, in the case of control predicates, both an alternative world and a modified variable assignment scope are passed when evaluating the abstracted node. Thus, sense abstraction is a theoretically coherent way of explaining the semantics of several different relative clause types that have previously been hard to connect.

In addition, sense abstraction offers an elegant way of writing a standard lexical denotation for *only*. While this is structurally quite different from the cases with subordinate clauses, we have demonstrated that the same +SENSE feature can be used to explain the semantics of both types of constructions.

References

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

[2] Frege, G. (1948). *Sense and reference*. The philosophical review, 57(3), 209-230.

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

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

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

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

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

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

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