An alternative semantics for presupposition*

Julian Grove

University of Rochester
julian.grove@gmail.com

Abstract

Charlow [3, 5, 4] develops a view of indefinites according to which they give rise to alternatives, an aspect of their meaning intended to explain the fact that they may generally take exceptional scope past scope islands. [5, 4] proposes that exceptional scope arises from the confluence of two factors: two polymorphic type shifts that regulate how the alternative sets denoted by indefinites compose with other expressions, and the availability of roll-up pied piping as a scope-taking strategy. The current paper explores a related observation about presupposition triggers: they, too, involve a kind of exceptional scope, which is reflected in their projection behavior. I propose a move analogous to that of [9] by treating definite noun phrases (and other presupposition triggers) on a par with indefinites, but within a variant of the alternative semantics introduced by Charlow. The resulting interpretation scheme preserves the alternative-style analysis of indefinites, but integrates a novel analysis of presupposition triggers that accounts for their flexible projection behavior. It additionally yields a simple analysis of presupposition accommodation according to which denotations for presupposition triggers are mapped onto those for indefinites.

1 The alternative view of exceptional scope

In recent work, Charlow [3, 5, 4] develops the view that indefinite noun phrases give rise to alternatives, and that this aspect of their meaning is responsible for their ability to take exceptional scope past the scope islands created by finite clause boundaries.

(1) If Theo has a brother, he'll bring a wetsuit that he owns.

(1) has two distinct readings. The first reading, which gives the indefinite wide scope, entails that Theo owns a wetsuit, and that he will bring it if he has a brother. The second reading, which gives the indefinite narrow scope, entails more weakly that Theo owns wetsuit if he has a brother. Charlow [5] analyzes such patterns as arising from the confluence of two factors: the indefinite's alternative-generating semantics and the availability of *roll-up pied piping* as a scope-taking strategy. According to the roll-up analysis, the indefinite first takes scope inside the consequent clause of the conditional; then, this clause itself takes scope above the conditional as a whole. This result is achieved by invoking two polymorphic type shifts, $\eta_{\mathtt{S}}$ and $\star_{\mathtt{S}}$, that can be interleaved in a semantic derivation with the rules of Functional Application and Predicate Abstraction [10]. These type shifts are defined in the following way, where '$\mathtt{S}\alpha$' abbreviates the type $\alpha \to t$ (it is the type of sets of alternative α's):

$$\eta_{\mathtt{S}} : \alpha \to \mathtt{S}\alpha \qquad\qquad (\star_{\mathtt{S}}) : \mathtt{S}\alpha \to (\alpha \to \mathtt{S}\beta) \to \mathtt{S}\beta$$

$$\eta_{\mathtt{S}}(v) = \{v\} \quad (= \lambda x.x = v) \qquad\qquad m \star_{\mathtt{S}} k = \bigcup_{x \in m} k(x) \quad (= \lambda y.\exists x : m(x) \wedge k(x)(y))$$

*Thanks to Dylan Bumford and Simon Charlow for helpful comments on the ideas presented here. Any errors are mine.

$$\eta(v) \star k = k(v) \qquad \text{(Left Identity)}$$
$$m \star \lambda x.\eta(x) = m \qquad \text{(Right Identity)}$$
$$(m \star n) \star o = m \star \lambda x.(n(x) \star o) \qquad \text{(Associativity)}$$

Figure 1: The monad laws

Together with these components, $\texttt{S}$, viewed as a type constructor, gives rise to a *monad*; that is, a map M from types to types associated with operators η_M and $\star_M$ having the type signatures

$$\eta_M : \alpha \to M\alpha$$
$$(\star_M) : M\alpha \to (\alpha \to M\beta) \to M\beta$$

and satisfying the laws in Figure 1. Charlow exploits this fact. For example, the law of Associativity allows roll-up pied piping to generate exceptional scope, since it means that whole-clause scope-taking can simulate the effect of the indefinite having taken scope above the conditional on its own in (1) (see [5] for discussion).

These points are illustrated in Appendix A, Figure 2, where exceptional scope for (1) is achieved. Here, the function `if` is defined as $\lambda\phi, \psi.\{\phi^{\natural} \to \psi^{\natural}\}$; following [5], it uses an *evaluation* function $(\cdot)^{\natural}$ turning a set of alternative truth values into a truth value:

$$(\cdot)^{\natural} : \texttt{S}t \to t$$
$$\phi^{\natural} = \exists p : p \in \phi \wedge p$$

`if` thus has the semantics of a material conditional while also evaluating its two arguments, effectively by checking that each of them returns $\top$ ('true') as once of its alternatives. Finally, the result of the derivation in Figure 2 is a set of alternative truth values; it evaluates via $(\cdot)^{\natural}$ to $\top$ just in case Theo owns a wetsuit which, if he has a brother, he'll bring.

The above is meant to give a gist of the workings of Charlow's system. The main goal of this paper is to slightly recast the alternative-monadic composition scheme, in order to countenance the semantic definedness conditions generated by presupposition triggers, along with the alternatives generated by indefinites. This is accomplished essentially by enriching the space inhabited by the functions characterizing alternative sets. While one could mix an analysis of presupposition triggers into an alternative semantics for indefinites in more than one way, I believe two considerations motivate the current move. First, as briefly discussed in the next section, presupposition triggers display exceptional scope behavior mirroring that displayed by indefinites; when presupposition triggers take scope, it is manifested not as non-determinism (as it is for indefinites), but as presupposition projection. This at least suggests that a monad should be available to countenance both.[1] Second, presupposition triggers, I argue, sometimes show up in a guise indicating that they should be analyzed as indefinites; that is, when they are locally accommodated [8]. In §4, it is shown that an alternative-style analysis of presupposition triggers makes a semantic analysis of presupposition accommodation straightforward to state.

[1]Such a monad could, in principle, arise from the application of a monad transformer to $\texttt{S}$, for example.

2 The exceptional scope of presupposition triggers

Remarkably, presupposition triggers appear to give rise to a pattern of projection behavior similar to the scopal behavior displayed by indefinites.

(2) If Theo has a brother, he'll bring his wetsuit.

(2) is most easily understood as presupposing that Theo has a wetsuit; but it may also be understood with the weaker presupposition *if Theo has a brother, he has a wetsuit*, given appropriate background information. This is illustrated in (3) (taken from [7]).

(3) We're going scuba diving later, and I don't know if Theo owns a wetsuit. But it seems that everyone who has a brother got a wetsuit for Christmas. So, if Theo has a brother, he'll bring his wetsuit.

Following [7], I assume that sentences with presupposition triggers are in general semantically ambiguous, though I won't further argue this point here (see [7] for a more developed discussion).

Geurts [6] points out that examples like (2) provide problems for satisfaction accounts of presupposition projection in the vein of Heim [8], since such accounts predict them to give rise to conditional presuppositions instead of unconditional ones. As illustrated in [7], a monadic account of presupposition that percolates undefined values, when used in tandem with roll-up pied piping, may supplement satisfaction accounts to achieve unconditional presuppositions for such examples. This strategy of explanation assimilates presupposition projection to scope-taking; global projection of a presupposition results from the presupposition trigger taking wide scope, while local projection results from narrow scope.

In the next section, I show how to treat presupposition projection in terms of an alternative semantics. In these terms, an analysis of the ambiguity of (2) may be wholly assimilated to an analysis of the ambiguity of (1). Indeed, both (in principle) display either the conditional inference that Theo owns a wetsuit if he has a brother, or the unconditional inference that he owns a wetsuit. What differs between the analyses of each is whether the relevant inference is hypothesized to arise by evaluating an alternative set encoding a truth condition or one encoding a definedness condition.

3 Alternatives and definedness at once

Heim [9] treats definites and indefinites as essentially similar in their semantic contributions, differing in the familiarity/novelty requirements they impose on the common ground. The move I make here is somewhat analogous, but it is taken within a variant of the Charlow-style semantics recalled in §1. It thus preserves the alternative-style analysis of indefinites; meanwhile, it fully integrates an analysis of presupposition triggers that regards them as giving rise to semantic definedness conditions.

As a first step, I adopt a trivalent semantics for truth values by introducing a type $t_\#$, inhabited by $\top$, $\bot$, along with a third "undefined" value $\#$. The type t, which is inhabited by only $\top$ and $\bot$, is thus a subtype of $t_\#$.[2] It is also useful to define a new binary connective $\wedge_\#$, as well as a new existential quantifier $\exists_\# : (\alpha \rightarrow t_\#) \rightarrow t_\#$, having the following semantics:

[2] The '$\top$' and '$\bot$' notation, and that for expressions of type t more generally, will equivocate between values of type t and those of type $t_\#$, a convention justified by viewing t as a subtype of $t_\#$. Throughout, it will be clear from context (e.g., a type annotation) which type is intended.

$\wedge_{\#}$	$\top$	$\bot$	$\#$
$\top$	$\top$	$\bot$	$\#$
$\bot$	$\bot$	$\bot$	$\#$
$\#$	$\#$	$\#$	$\#$

$$\exists_{\#} x : \phi(x) = \begin{cases} \top & \exists x : (\phi(x) = \top) \\ \bot & \exists x : (\phi(x) = \bot) \wedge \neg\exists x : (\phi(x) = \top) \\ \# & \forall x : (\phi(x) = \#) \end{cases}$$

The connective $\wedge_{\#}$ effectively has a weak Kleene semantics, percolating up undefinedness from either of its conjuncts. Meanwhile, $\exists_{\#}$ is highly charitable as an existential quantifier, resulting in an undefined value only if its scope is undefined for all possible values of the quantified variable. (Otherwise, it acts similar to a classical existential quantifier.) Finally, it is useful to introduce an operator $\delta : t \to t_{\#}$ akin to the δ-operator of [2]; that is, $\delta(\top) = \top$ and $\delta(\bot) = \#$.

Given these pieces, one may redefine the operators from the first section, including the evaluation function, with the new type $t_{\#}$ in place of t. Characterizing sets as functions of type $\alpha \to t_{\#}$ instead of functions of type $\alpha \to t$ means that membership in an alternative "set" may be *undefined*; that is, where the value of its characteristic function is $\#$. Here, '$\mathtt{S}_{\#}\alpha$' abbreviates the type $\alpha \to t_{\#}$.

$$\eta_{\mathtt{S}_{\#}} : \alpha \to \mathtt{S}_{\#}\alpha \qquad (\star_{\mathtt{S}_{\#}}) : \mathtt{S}_{\#}\alpha \to (\alpha \to \mathtt{S}_{\#}\beta) \to \mathtt{S}_{\#}\beta \qquad (\cdot)^{\lightning_{\#}} : \mathtt{S}_{\#}t_{\#} \to t_{\#}$$

$$\eta_{\mathtt{S}_{\#}}(v) = \lambda x.\delta(x = v) \qquad m \star_{\mathtt{S}_{\#}} k = \lambda y.\exists_{\#} x : m(x) \wedge_{\#} k(x)(y) \qquad \phi^{\lightning_{\#}} = \exists_{\#} p : \phi(p) \wedge_{\#} p$$

This encoding of alternatives also gives rise to a monad (Theorem 2 of Appendix C). As a consequence, the exceptional scope behavior characteristic of monadic semantics carries over.

Using $\mathtt{S}_{\#}$ in place of $\mathtt{S}$ allows for the derivation of (2) in Figure 5 of Appendix B. Here, *his wetsuit* is taken to denote $\lambda x.\delta(\mathsf{suitOfT}(x))$ — that is, the characteristic function of a "set" whose value is $\top$ on Theo's wetsuit (if it exists) and $\#$ everywhere else. The meaning of *if* in this derivation ($\mathtt{if}_{\#}$) is taken to be the function $\lambda\phi, \psi.\eta_{\mathtt{S}_{\#}}(\phi^{\lightning_{\#}} \Rightarrow \psi^{\lightning_{\#}})$ of type $\mathtt{S}_{\#}t_{\#} \to \mathtt{S}_{\#}t_{\#} \to \mathtt{S}_{\#}t_{\#}$. The connective $\Rightarrow$ featured here takes two values of type $t_{\#}$ onto a third and is assumed to have a semantics in the style of [12]:

$\Rightarrow$	$\top$	$\bot$	$\#$
$\top$	$\top$	$\bot$	$\#$
$\bot$	$\top$	$\top$	$\top$
$\#$	$\#$	$\#$	$\#$

Given either $\top$ or $\bot$ as arguments, $\Rightarrow$ behaves as a material conditional. Otherwise, it automatically projects undefinedness in its left conjunct, while undefinedness in its right conjunct is projected only if the left conjunct is $\top$.

Analogous to the derivation of (1), the type resulting for (2) is $\mathtt{S}_{\#}t_{\#}$: that of a "set" of truth values (here of type $t_{\#}$), but, in this case, whose membership relation may not always be defined. If this set is evaluated using $(\cdot)^{\lightning_{\#}}$, a value of type $t_{\#}$ is produced. The representation of this value is quite complex, given the term decorating the root in Figure 5, but it may be simplified by relying on the theorem δ-elim of Appendix C:

$$
\begin{aligned}
& (\lambda p.\exists_{\#}\phi, x : \delta(\mathsf{suitOfT}(x)) \wedge_{\#} \delta(\phi = \eta_{\mathsf{S}_{\#}}(\mathsf{bring}(x)(\mathsf{t}))) \wedge_{\#} \delta(p = ((\eta_{\mathsf{S}_{\#}}(\mathsf{bro}(\mathsf{t})))^{\lightning\#} \Rightarrow \phi^{\lightning\#})))^{\lightning\#} \\
= \; & (\lambda p.\exists_{\#}\phi, x : \delta(\mathsf{suitOfT}(x)) \wedge_{\#} \delta(\phi = \eta_{\mathsf{S}_{\#}}(\mathsf{bring}(x)(\mathsf{t}))) \wedge_{\#} \delta(p = ((\exists_{\#} q : \delta(q = \mathsf{bro}(\mathsf{t})) \wedge_{\#} q) \Rightarrow \phi^{\lightning\#})))^{\lightning\#} \\
& \hfill \text{(by defs. of } (\cdot)^{\lightning\#} \text{ and } \eta_{\mathsf{S}_{\#}}) \\
= \; & (\lambda p.\exists_{\#} x : \delta(\mathsf{suitOfT}(x)) \wedge_{\#} \delta(p = (\mathsf{bro}(\mathsf{t}) \Rightarrow (\eta_{\mathsf{S}_{\#}}(\mathsf{bring}(x)(\mathsf{t})))^{\lightning\#})))^{\lightning\#} \quad \text{(by } \delta\text{-elim)} \\
= \; & (\lambda p.\exists_{\#} x : \delta(\mathsf{suitOfT}(x)) \wedge_{\#} \delta(p = (\mathsf{bro}(\mathsf{t}) \Rightarrow \exists_{\#} q : \delta(q = \mathsf{bring}(x)(\mathsf{t})) \wedge_{\#} q)))^{\lightning\#} \\
& \hfill \text{(by defs. of } (\cdot)^{\lightning\#} \text{ and } \eta_{\mathsf{S}_{\#}}) \\
= \; & (\lambda p.\exists_{\#} x : \delta(\mathsf{suitOfT}(x)) \wedge_{\#} \delta(p = (\mathsf{bro}(\mathsf{t}) \Rightarrow \mathsf{bring}(x)(\mathsf{t}))))^{\lightning\#} \quad \text{(by } \delta\text{-elim)} \\
= \; & \exists_{\#} p, x : \delta(\mathsf{suitOfT}(x)) \wedge_{\#} \delta(p = (\mathsf{bro}(\mathsf{t}) \Rightarrow \mathsf{bring}(x)(\mathsf{t}))) \wedge_{\#} p \quad \text{(by def. of } (\cdot)^{\lightning\#}) \\
= \; & \exists_{\#} x : \delta(\mathsf{suitOfT}(x)) \wedge_{\#} (\mathsf{bro}(\mathsf{t}) \Rightarrow \mathsf{bring}(x)(\mathsf{t})) \quad \text{(by } \delta\text{-elim)}
\end{aligned}
$$

The semantics of $\exists_{\#}$ dictates that the value that results here is equal to $\#$ if Theo has no wetsuit, and to $\top$ or $\bot$ otherwise, as according to whether or not he will bring his wetsuit if he has a brother. Thus the presupposition predicted for (2) is that Theo has a wetsuit. Though it isn't provided here, the derivation of the conditional presupposition that Theo has a wetsuit *if he has a brother* can be generated for (2) by giving the presupposition trigger narrow scope — that is, by invoking $\eta_{\mathsf{S}_{\#}}$ one fewer time in the scope of the moved trigger and evaluating the consequent of the conditional *in situ*.

Meanwhile, from a monadic bird's-eye view, the analysis of (1) is almost unchanged. Figure 4 of Appendix B gives a derivation for (1) using $\mathsf{S}_{\#}$ which is equivalent to that given in Figure 2, save the details that (1) the monadic operators have been redefined, and (2) formulae that were of type t before are now of type $t_{\#}$. Evaluating the result with $(\cdot)^{\lightning\#}$ yields a value whose representation can be simplified, as above, to yield:

$$\exists_{\#} x : \mathsf{suitOfT}(x) \wedge_{\#} (\mathsf{bro}(\mathsf{t}) \Rightarrow \mathsf{bring}(x)(\mathsf{t}))$$

The value of this formula is either $\top$ or $\bot$: $\top$ if Theo has a wetsuit that he will bring if he has a brother, and $\bot$ if not. What was a definedness condition in the analysis of (2) is a here a truth condition.

4 Accommodation

The combined alternative semantics treats indefiniteness and presupposition uniformly, so that indefinite expressions and presupposition triggers are of the same semantic types $\mathsf{S}_{\#}\alpha$ (for some α) and thus denote in the same space of functions into $t_{\#}$. The only semantic difference between the two kinds of expression, from the current perspective, is the way in which they use this space. Indefinites encode alternatives in terms of the values $\top$ and $\bot$: a live alternative is mapped to $\top$, while any non-alternative is mapped to $\bot$. Meanwhile, presupposition triggers encode presuppositions in terms of the values $\top$ and $\#$: a value that *satisfies* an expression's presupposition is mapped to $\top$, while any value which doesn't is mapped to $\#$. When values of type $\mathsf{S}_{\#}t_{\#}$ are evaluated via $(\cdot)^{\lightning\#}$, these encodings are cashed out as existential truth conditions for indefiniteness and existential definedness conditions for presupposition triggers.

Treating the two as akin allows for a straightforward implementation of the phenomenon known as "local accommodation" [8], a species of the accommodation process described by

Lewis [11], but which appears inside the scope of linguistic operators, e.g., negation. We can look at Heim's original example of the phenomenon to illustrate it.

(4) The king of France didn't come.

Consider (as Heim does) a scenario in which (4) is uttered and then followed up with *because France doesn't have a king*. In such a context, (4) may be understood as denying that France has a king who came. In the spirit of [3], negation may be analyzed in terms of an operator `not` which cancels the alternatives generated by indefinites:

$$\texttt{not} : \mathtt{S}_{\#}t_{\#} \to t_{\#} \qquad \qquad \neg\top = \bot$$
$$\texttt{not}(\phi) = \neg(\phi^{\natural\#}) \qquad \text{where} \qquad \neg\bot = \top$$
$$\neg\# = \#$$

Note that the ¬ connective leaves # unmodified. As a result, `not` will allow presuppositions (though not alternatives) to project past it. Finally, we may introduce a polymorphic operator `accom` which has the effect of turning definedness conditions into truth conditions:

$$\texttt{accom} : \mathtt{S}_{\#}\alpha \to \mathtt{S}_{\#}\alpha \qquad \qquad \delta^{-1}(\top) = \top$$
$$\texttt{accom}(m) = \lambda x.\delta^{-1}(m(x)) \qquad \text{where} \qquad \delta^{-1}(\bot) = \bot$$
$$\delta^{-1}(\#) = \bot$$

`accom` relies on an operator δ^{-1}, which is effectively a left inverse of δ: it leaves $\top$ and $\bot$ unmodified, but takes # back onto $\bot$.[3]

Though a lack of space precludes a full derivation, we may use these components to provide a meaning representation for (4) which is suggestive of one:

$$\begin{aligned} & \texttt{not}(\texttt{accom}(\lambda p.\exists_{\#}x : \delta(\mathsf{kof}(x)) \wedge_{\#} \delta(p = \mathsf{come}(x)))) \\ = \ & \texttt{not}(\lambda p.\delta^{-1}(\exists_{\#}x : \delta(\mathsf{kof}(x)) \wedge_{\#} \delta(p = \mathsf{come}(x)))) && \text{(by def. of } \texttt{accom}) \\ = \ & \neg(\exists_{\#}p : (\delta^{-1}(\exists_{\#}x : \delta(\mathsf{kof}(x)) \wedge_{\#} \delta(p = \mathsf{come}(x))) \wedge_{\#} p) && \text{(by def. of } \texttt{not}) \end{aligned}$$

By attending to the possible values of the formula serving as the argument to δ^{-1} here (and with the help of δ-elim), it is possible to see that this formula is equivalent to the following one:

$$\neg\exists_{\#}x : \mathsf{kof}(x) \wedge_{\#} \mathsf{come}(x)$$

Thus by first locally accommodating the presupposition of (4) via `accom`, we have managed to cancel it via `not`.

5 Conclusion

An alternative semantics encoding presuppositional definedness conditions allows an explanation of the semantic kinship between indefinites and definites (and presupposition triggers, generally) manifest in their similar scopal behavior and when presupposition triggers are accommodated. On the alternative analysis of presupposition, both give rise to some kind of existence inference: indefinites require the existence of an alternative (where their corresponding inference is *at issue*), while presupposition triggers have a similar requirement, though the stakes are about definedness rather than truth. Future work will have to determine how plausibly these assumptions can be extended to the variety of presupposition triggers that exist.

[3] It is thus akin to the A operator of [1].

A Exceptional scope via S

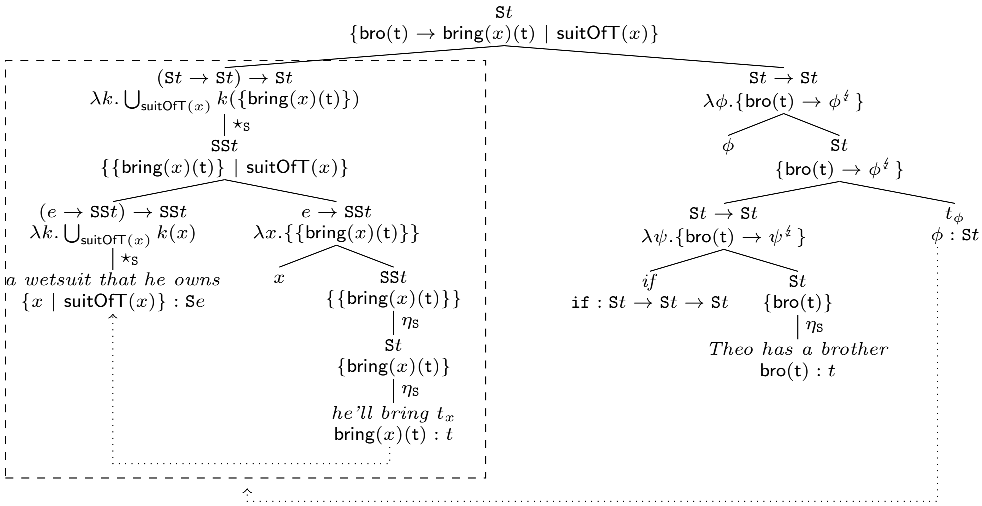

Figure 2: Deriving (1) via S

B Exceptional scope via $S_\#$

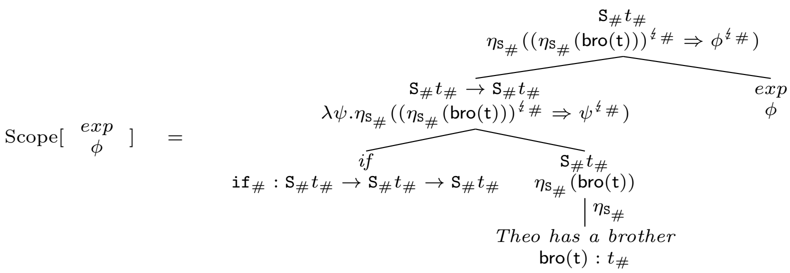

Figure 3: Scope of the moved expression in Figures 4 and 5

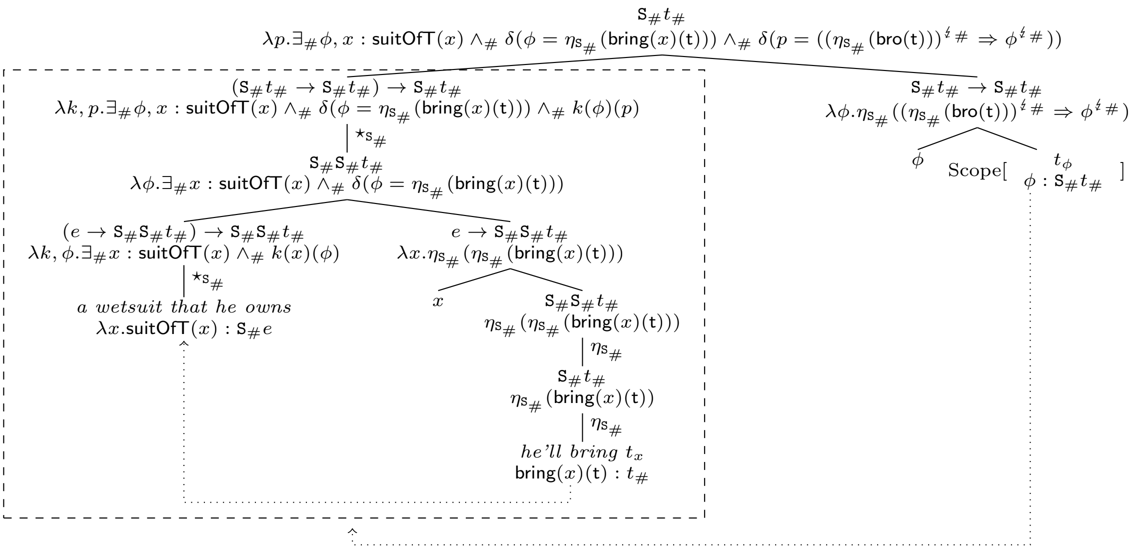

Figure 4: Deriving (1) via $\mathtt{S}_\#$

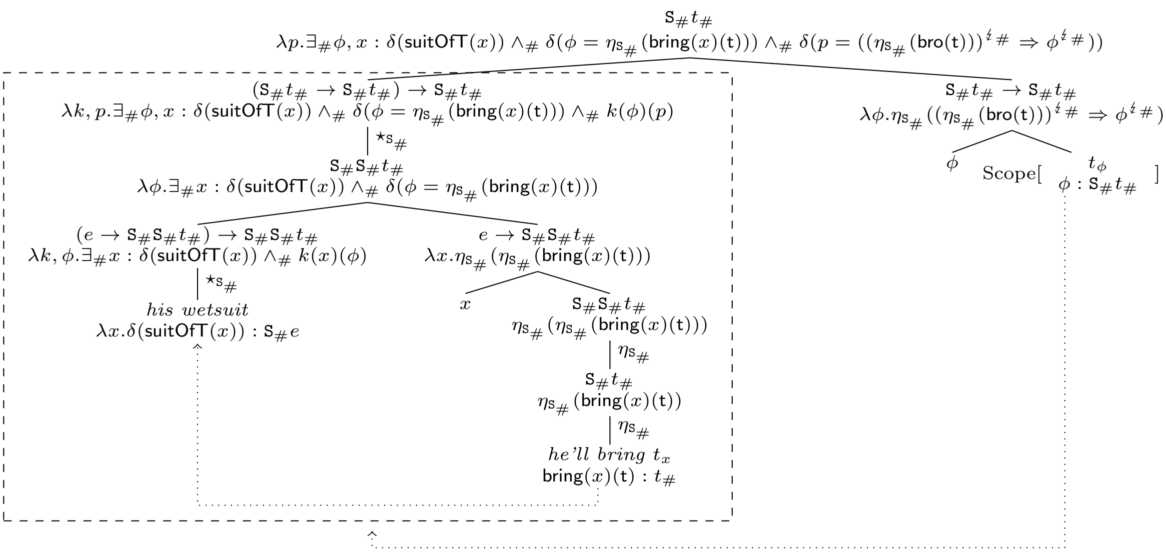

Figure 5: Deriving (2) via $\mathtt{S}_\#$

C Proof that $\mathtt{S}_\#$ is a monad

This section offers only proof sketches. The actual proofs are available at http://github.com/juliangrove/trivalent-alternatives.

The following theorem is useful for simplifying formulae that result from the composition of monadic operators.

Theorem 1 (δ-elim)**.** $\exists_\# x : \delta(x = v) \wedge_\# \phi(x) \quad = \quad \phi(v)$.

Proof. Both the LHS and RHS are either true, false, or undefined. If the LHS is true, then there is a witness to the existential quantifier which, when chosen as the value of x, makes its scope true; this witness must be v, making the RHS true. If the LHS is false, then choosing v as the value of x makes the quantifier's scope false, in turn making the RHS false. If the LHS is undefined, then choosing v as the value of x makes the quantifier's scope undefined, in turn making the RHS undefined. □

Theorem 2. $\mathtt{S}_\#$ is a monad.

Proof.
Left Identity

$$\begin{aligned} \eta_{\mathtt{S}_\#}(v) \star_{\mathtt{S}_\#} k &= \lambda y.\exists_\# x : \delta(x = v) \wedge k(x)(y) && \text{(by def., } \equiv_\beta) \\ &= \lambda y.k(v)(y) && \text{(by } \delta\text{-elim)} \\ &= k(v) && \text{(by } \equiv_\eta) \end{aligned}$$

Right Identity
This relies on $\wedge_\#$ being commutative.

$$\begin{aligned} m \star_{\mathtt{S}_\#} \lambda x.\eta_{\mathtt{S}_\#}(x) &= \lambda y.\exists_\# x : m(x) \wedge_\# \delta(y = x) && \text{(by def., } \equiv_\beta) \\ &= \lambda y.m(y) && \text{(by commutativity of } \wedge_\#, \delta\text{-elim)} \\ &= m && \text{(by } \equiv_\eta) \end{aligned}$$

Associativity
This relies on $\exists_\#$ commuting past $\wedge_\#$ when it binds a variable in one conjunt, $\exists_\#$ commuting with itself, and the associativity of $\wedge_\#$.

$$\begin{aligned} (m \star_{\mathtt{S}_\#} n) \star_{\mathtt{S}_\#} o &= \lambda z.\exists_\# y : (m \star_{\mathtt{S}_\#} n)(y) \wedge_\# o(y)(z) && \text{(by def.)} \\ &= \lambda z.\exists_\# y : (\exists_\# x : m(x) \wedge_\# n(x)(y)) \wedge_\# o(y)(z) && \text{(by def., } \equiv_\beta) \\ &= \lambda z.\exists_\# y : \exists_\# x : (m(x) \wedge_\# n(x)(y)) \wedge_\# o(y)(z) && \text{(by comm. of } \exists_\#/\wedge_\#) \\ &= \lambda z.\exists_\# x : \exists_\# y : (m(x) \wedge_\# n(x)(y)) \wedge_\# o(y)(z) && \text{(by commutativity of } \exists_\#) \\ &= \lambda z.\exists_\# x : \exists_\# y : m(x) \wedge_\# (n(x)(y) \wedge_\# o(y)(z) && \text{(by associativity of } \wedge_\#) \\ &= \lambda z.\exists_\# x : m(x) \wedge_\# \exists_\# y : n(x)(y) \wedge_\# o(y)(z) && \text{(by comm. of } \exists_\#/\wedge_\#) \\ &= \lambda z.\exists_\# x : m(x) \wedge_\# (n(x) \star_{\mathtt{S}_\#} o)(z) && \text{(by } \equiv_\beta\text{, def.)} \\ &= m \star_{\mathtt{S}_\#} \lambda x.(n(x) \star_{\mathtt{S}_\#} o) && \text{(by } \equiv_\beta\text{, def.)} \end{aligned}$$

□

References

[1] David Beaver and Emiel Krahmer. A Partial Account of Presupposition Projection. *Journal of Logic, Language and Information*, 10(2):147–182, 2001. URL: https://doi.org/10.1023/A:1008371413822.

[2] David I. Beaver. The kinematics of presupposition. In Paul Dekker and Martin Stokhof, editors, *Proceedings of the Eighth Amsterdam Colloquium*, pages 17–36, Amsterdam. ILLC, 1992.

[3] Simon Charlow. *On the semantics of exceptional scope*. PhD Thesis, New York University, New York, 2014. URL: https://semanticsarchive.net/Archive/2JmMWRjY.

[4] Simon Charlow. Static and dynamic exceptional scope. Publisher: Rutgers University Published: LingBuzz, 2020. URL: https://ling.auf.net/lingbuzz/004650.

[5] Simon Charlow. The scope of alternatives: indefiniteness and islands. en. *Linguistics and Philosophy*, 43(4):427–472, 2020. ISSN: 1573-0549. DOI: 10.1007/s10988-019-09278-3.

[6] Bart Geurts. Local Satisfaction Guaranteed: A Presupposition Theory and Its Problems. *Linguistics and Philosophy*, 19(3):259–294, 1996. DOI: 10.1007/BF00628201.

[7] Julian Grove. Presupposition projection as a scope phenomenon. en. *Semantics and Pragmatics*, 15(15), December 2022. ISSN: 1937-8912. DOI: 10.3765/sp.15.15.

[8] Irene Heim. On the Projection Problem for Presuppositions. In Michael D. Barlow, Daniel P. Flickinger, and Michael Westcoat, editors, *The West Coast Conference on Formal Linguistics (WCCFL)*, volume 2, pages 114–125, Stanford. Stanford University Press, 1983.

[9] Irene Heim. *The Semantics of Definite and Indefinite Noun Phrases*. PhD Thesis, University of Massachussetts, Amherst, 1982. URL: https://semanticsarchive.net/Archive/Tk0ZmYyY/.

[10] Irene Heim and Angelika Kratzer. *Semantics in Generative Grammar*. Blackwell, Malden, MA, 1998.

[11] David Lewis. Scorekeeping in a Language Game. In Rainer Bäurele and Arnim von Stechow, editors, *Semantics from Different Points of View*. Volume 6, Springer Series in Language and Communication, pages 172–187. Springer, Berlin, 1979. URL: https://doi.org/10.1007/978-3-642-67458-7_12.

[12] Stanley Peters. A truth-conditional formulation of Karttunen's account of presupposition. *Synthese*, 40(2):301–316, 1979. DOI: 10.1007/BF00485682.