Deriving the (a)-symmetries of presupposition projection*

Alexandros Kalomoiros

University of Pennsylvania

1. Introduction

This paper is concerned with the projection problem for presuppositions, arguing for a novel projection algorithm that derives asymmetric conjunction, but symmetric disjunction through a single mechanism.

Certain lexical items impose conditions (presuppositions) on the context in which they are uttered:

(1) a. *Context:* We do not know whether or not John used to have research interests in Tolkien.
  b. #John continues to have research interests in Tolkien.

'Continues to have research interests in Tolkien' presupposes 'used to have research interests in Tolkien', hence infelicity arises when (1b) is uttered in a (global) context that does not entail this presupposition.

The projection problem enters the stage when presuppositions are embedded under various operators:

(2) *Projection Problem:* How are the presuppositions of a complex sentence derived from the presuppositions of its parts? (Langendoen and Savin 1971 Karttunen 1973, Heim 1983, and much subsequent work).

To see the motivation for the problem consider the following sentences:

(3) a. *Context:* We do not know whether or not John used to have research interests in Tolkien.

*Thanks to Florian Schwarz, Julie Legate, Anna Papafragou, Martin Salzmann, Phillipe Schlenker, Matt Mandelkern, Muffy Siegel, Andrea Beltrama, Lefteris Paparounas, Ryan Budnick and Spencer Caplan for useful feedback and suggestions. Thanks also to the NELS 52 audiences. All errors are my own.

b. #John continues to have research interests in Tolkien and he used to have interests in Tolkien.
c. John used to have research interests in Tolkien and he continues to have research interests in Tolkien.

Example (3b) is felt to carry the presupposition that 'John used to have research interests in Tolkien'; in this case we say that the presupposition *projects*. Hence, when (3b) is uttered in a context where we have ignorance about whether or not John used to have such research interests, infelicity arises.

On the other hand, the conjunction in (3c) is not felt to presuppose that 'John used to have research interests in Tolkien' (although it entails it). In this case we say that the presupposition is *filtered* (in the terminology of Karttunen 1973). The only difference between (3b) and (3c) is the order of the conjuncts: in (3b) the presupposition-bearing conjunct comes first, while in (3c) it comes second (see also Mandelkern et al. (2020) on experiments that confirm this difference for conjunction).

Thus, conjunction represents a case where the filtering of presuppositions is asymmetric: information in the left conjunct can filter a presupposition in the right conjunct, but not the other way around. An intuitive explanation for this is that the the left conjunct in (3c) is evaluated first (Stalnaker 1974), just by virtue of the fact that one encounters it first as the sentence unfolds in time. Thus the information carried by it is integrated in the context before the second conjunct is evaluated. In this way, the second conjunct is evaluated against a context that entails its presupposition. That's why no projection arises.

Conversely, in (3b) the first conjunct is evaluated in a context which does not entail the presupposition that John used to have research interests in Tolkien. That information only comes later, in the second conjunct, and is not in principle accessible for the evaluation of the first conjunct.

While conjunction behaves asymmetrically, this is not universally true across connectives. A famous example is disjunction. Presuppositions in the second disjunct of a disjunction are filtered if the negation of the first disjunct entails the presupposition, (Karttunen 1973):

(4) Either John never had research interests in Tolkien or he continues to have such research interests.

However, reversing the disjuncts does not change the filtering pattern (Partee's 'bathroom sentences'):[1]

(5) Either John continues to have research interests in Tolkien or he never had such interests.

[1] The reason these disjunction are known as 'bathroom sentences' is that Partee's original examples were of the following form:

(i) Either the bathroom is in a weird place or this house has no bathroom.

Neither (4) nor (5) are felt to presuppose that John used to have research interests in Tolkien. This raises the question of how to best model these (a)-symmetries in projection: projection from conjunction would appear to imply that the mechanism determining filtering is fundamentally asymmetric, with the asymmetry perhaps rooted in the left-to-right incrementality inherent in the way we parse a sentence as it unfolds in time. Disjunction though appears to provide an argument for a symmetric filtering mechanism. The question then is whether we need two different filtering mechanisms, one symmetric and one asymmetric, to account for this landscape (as proposed e.g., by Schlenker (2008, 2009), Rothschild (2011)); or whether it is possible to unify the phenomena as instances of one general filtering algorithm.

The rest of this paper is organised as follows: Section 2 briefly reviews Schlenker's (2009) theory of local contexts, which aims to tackle the problem in parsing-oriented framework, by having two filtering mechanisms, one symmetric and one asymmetric. We argue that recent experimental evidence points to the existence of a single filtering mechanism. Section 3 takes on the task of building such a single mechanism (the system is dubbed *Limited Symmetry*) within the a parsing-oriented bivalent framework, and shows how it derives asymmetry for conjunction, but symmetry for disjunction. Section 4 concludes.

2. Schlenker 2009

We give a brief introduction to Sclenker's approach to the projection problem (Schlenker 2009), as our own solution in section 3 follows the spirit of this approach. Schlenker provides a reconstruction of Kartunnen's notion of 'local context' (Karttunen 1974), whereby *a presupposition of sentence S must be entailed by its local context.*

At the core of Schlenker's proposal is the idea that in determining what counts as a local context, there's an underlying strategy of only evaluating presuppositions relative to those possible worlds in which the truth value of the complex sentence overall is not already determined by other parts of the sentence. How precisely this plays out depends on the truth-functional properties of the connective in question, which ultimately account for differences in local contexts, e.g., with conjunction involving consideration of information of another conjunct, whereas disjunction requires consideration of the negation of another disjunct.

Schlenker assumes a simple language with a classical bivalent semantics. Here, we focus on the propositional fragment of this language (see section 3.2 for a statement of this fragment). Following Stalnaker, a context C is modelled as a set of possible worlds (Stalnaker 1974). The notation $C \models p$ means that the proposition expressed by p is True in every world in C. Here's the definition for the asymmetric local context of an sentence S:

Definition 1 Asymmetric Local Context:[2] The asymmetric local context of a sentence S in a syntactic environment $a\ \_\ b$ and global context C, is the strongest proposition r such that for all sentences D and good finals b', $C \models a(r\ and\ D)b' \leftrightarrow a(D)b'$

[2]This definition focuses on the propositional case and is borrowed from Mandelkern and Romoli (2017). See Schlenker (2009) for full definitions generalized to a more expressive language.

Now consider a conjunction like $(p\ and\ q)$, and say we want to calculate the local context for q in a global context C. Applying the definition, we need to calculate the strongest proposition r such that for all sentences D and good finals b', $C \models (p\ and\ (r\ and\ D)b' \leftrightarrow (p\ and\ (D)b'$. The only possible good final here is a left parenthesis,). One proposition that does the job is the proposition expressed by p: $C \models (p\ and\ (p\ and\ D)b' \leftrightarrow (p\ and\ (D)b'$. To show that p is the strongest proposition that we could conjoin here, suppose that there is a proposition r that excludes a C-world w' that satisfies p, i.e., r is False in w'. Suppose also that D is true in w'. In this case, $(p\ and\ D)$ is True in w', but $(p\ and\ (r\ and\ D))$ is False; so the equivalence $C \models (p\ and\ D) \leftrightarrow (p\ and\ (r\ and\ D))$ fails. Therefore, the local context for a second conjunct is the first conjunct.

Applying similar reasoning, one can show that the local context for the second disjunct q of a disjunction $(p\ or\ q)$ is $(not\ p)$: $C \models (p\ or\ (D)b' \leftrightarrow (p\ or\ ((not\ p)\ and\ D)b'$[3]. As in the conjunction case above, setting the local context to something stronger than $(not\ p)$ leads to failure of the desired equivalence. So, the local context of a second disjunct is the negation of the first disjunct, thus correctly predicting filtering in cases like (4).

So far, this system predicts correct filtering conditions for presuppositions in the second conjunct/disjunct. However, it predicts the same filtering conditions for presuppositions in the first conjunct/disjunct, as in both cases the asymmetric local context is the global context. The reason for this is that in both cases we are calculating the strongest r that can be conjoined to a first conjunct/disjunct D such that $C \models ((D)b' \leftrightarrow ((r\ and\ D)b'$, for all D, b'. It does not matter what the connective is, as the connective essentially 'hides' in b', which quantifies over all possible good finals. Therefore, the only possible restriction of the first conjunct/disjunct is $r = C$ (which is no restriction contextually). If we try to conjoin something stronger than C, then the desired equivalence fails. Thus, a presupposition in the first conjunct/disjunct projects if it is not entailed by C.

The result about first disjuncts/conjuncts seems to be contradicted by cases like (5), where the second disjunct is apparently filtering the presupposition of the first disjunct. To account for this, Schlenker (2009) proposes a symmetric definition of local contexts:

Definition 2 Symmetric Local Context: The symmetric local context of a sentence S in a syntactic environment $a\ \_\ b$ and global context C is the strongest proposition r such that for all sentences D, $C \models a(r\ and\ D)b \leftrightarrow a(D)b$.

Now we are no longer quantifying over all possible good finals, but rather allow access to the actual sentence completion b. This allows the actual continuation of the first disjunct (namely, *or* q)) to be taken into account. Applying similar reasoning as in the case of the asymmetric local context of a second disjunct, the symmetric local context of a first disjunct is $(not\ q)$, which is what we need here. This definition also predicts a symmetric local context for a first conjunct, namely the second conjunct, which predicts cases of symmetric filtering in conjunction. To account for the fact that symmetric filtering in conjunction seems much rarer, Schlenker (2009) posits that asymmetric local contexts are the default (it is in this sense that the system is parsing-oriented), while accessing symmetric local contexts involves overriding this default and hence carries a processing cost.

[3] $(p \vee q) \leftrightarrow (p \vee ((\neg p) \wedge q))$ is a tautology.

Nevertheless, recent experimental evidence casts doubt on this claim. Mandelkern et al. (2020) show that it is not possible to prevent a presupposition projecting from the first conjunct, even in situations where the only way to prevent infelicity would be to access a symmetric mechanism. Moreover, the idea that symmetric filtering is costly makes a prediction that disjunctions like (4) where the presupposition is in the second disjunct are more felicitous than disjunction like (5), where the presupposition is in the first disjunct. Kalomoiros and Schwarz (2021) tested this prediction by adapting the Mandelkern et al. design to disjunctions: the results showed no asymmetry between the two types of disjunction; both were equally felicitous (see the cited papers for more details on the experiments). Therefore, it seems preferable to derive the projection (a)-symmetries via a single mechanism.

3. Limited Symmetry

3.1 Informal Idea

We develop a parsing-oriented system, which we dub *Limited Symmetry*, inspired by the approach of Schlenker (2009), but where the asymmetry of conjunction and the symmetry of disjunction follow from a single mechanism. We start with an informal version of the system; the formal definitions and main results are in sections 3.2, and 3.3.

Recall that on the asymmetric conception of local contexts, in calculating the local context of a first conjunct/disjunct, the parser has no access to the connective; this is part of the reason why the system does not differentiate first conjuncts vs first disjucnts. To differentiate between them, we need our projection algorithm to be able to have access to the $($*p and*, $($*p or* substrings of a conjunction and a disjunction (i.e., access to the connectives).

To do this we will assume that a conjunction like $(p\ and\ q)$ is parsed as $[(,\ (p,\ (p\ and,\ (p\ and\ q,\ (p\ and\ q)]$, whereas a disjunction of the form $(p\ or\ q)$ is parsed as $[(,\ (p,\ (p\ or,\ (p\ or\ q,\ (p\ or\ q)]$. Each entry in these lists represents a parsing step for the given sentence. Here is now the core intuition: at each parsing step, the parser is attempting to calculate all the sets of worlds where the truth value of the whole sentence is already determined (as either True or False) for all possible sentence completions (good finals). The aim is to update, i.e., to get rid of the worlds in the global context C where the sentence is False as fast as possible, while keeping the worlds where it is True. At each parsing step, the set of worlds where the sentence is already False is removed from the global context C and the update process restarts with $C' = C - \{w|\ the\ sentence\ is\ already\ False\ in\ w\}$ as the new global context.

Atomic sentences that carry presuppositions will be notated as $p'p$, where p' represents the presuppositonal component and p the non-presuppositional, assertive component; the semantics of such expressions is conjunctive: $p'p$ is True iff both p' and p are True (this is borrowed from Schlenker 2009).

If in parsing a sentence S, at some point we encounter a $p'p$ sub-sentence, and if we can calculate either a set of worlds where S is already True or a set of worlds where S is already False at that parsing step, a constraint applies: presuppositions need to be ‘informationally null' (this is basically the classic condition that a presupposition cannot introduce new information).

More formally, we require that no matter what the assertive component p of $p'p$ is, all of the worlds where S is already True/False must be worlds where $(S)^-$ is already True/False, with $(S)^-$ being a version of S where p' has been deleted. If this holds, then the update continues. If it doesn't, then the update process exits (presupposition failure).

Here's how this system breaks the symmetry between conjunction and disjunction: in a conjunction $(r'r\ and\ q)$, at parsing step $(r'r\ and$, we already know that the sentence is False in all C-worlds where $r'r$ is False. Therefore, it is at this point that the presupposition requirement has a chance to create trouble; the set of worlds where the conjunction is already False no matter the continuation is $\{w|r'(w)=0\ or\ r(w)=0\}$ (recall $r'r$ is interpreted as conjunction). We need to check if the presupposition r' is informationally null with regards to this set, no matter the assertive component r. So, we need to check if for any r:

(6) $\{w\in C|r'(w)=0\ or\ r(w)=0\}\subseteq\{w\in C|r(w)=0\}$

This will hold in C just in case $\{w\in C|r'(w)=0\}=\emptyset$, i.e., iff $C\models r'$ (see also **Fact 3**).

So, r' projects unless entailed by the context. Conversely, given a disjunction $(r'r\ or\ q)$, at parsing step $(r'r\ or$, we can only determine a set of worlds where the disjunction is True for all possible continuations, namely all the worlds where $r'r$ is True: $\{w\in C|\ r'(w)=1\ and\ r(w)=1\}$, which is a subset of $\{w\in C|\ r(w)=1\}$ (the corresponding set if we only consider the assertive part), for any r . This already shows that conjunction and disjunction have different points where the projection requirement can create trouble (see **Fact 5** for how exactly symmetry comes about). We now turn to formalizing *Limited Symmetry*.

3.2 Formalization

3.2.1 Language and Semantics

We restrict ourselves to a propositional language $\mathscr{L}$ (inspired by Schlenker (2009)):

(7) $\phi := p_i\mid p'_j p_k\mid(not\ \phi)\mid(\phi\ and\ \ \phi)\mid(\phi\ or\ \phi)\mid(if\ \phi.\ \phi)$ $\qquad i,j,k\in\mathbb{N}$

In $p'_j p_k$, p'_j is meant to be understood as the entailments that have been marked as presuppositional and p_k as the non-presuppositional entailments. Below, we will omit subscripts and will be using lower case letters to name propositions $(p,q,r,\ldots$ etc.)

The intended models of this language are pairs $\langle W,\ I\rangle$, where W is a set of worlds, and I is a function assigning to each propositional constant of $\mathscr{L}$ a set of worlds. Our semantics is bivalent and follows the standard truth tables. Sentences that carry presuppositional entailments are treated like conjunctions (following Schlenker 2009):

Definition 3: Truth in a world

- p is T in w iff $w\in I(p)$
- $p'p$ is T in w iff $w\in I(p')$ and $w\in I(p)$

- $(not\ \phi)$ is T in w iff ϕ is F in w
- $(\phi\ and\ \psi)$ is T in w iff ϕ is T in w and ψ is T in w
- $(\phi\ or\ \psi)$ is T in w iff ϕ is T in w or ψ is T in w
- $(if\ \phi.\ \psi)$ is T in w iff ϕ F in w or ψ is T in w

We follow Schlenker (2009) in taking a sentence S to be evaluated against a context C (the global context), where C is a set of worlds (intuitively, the set of worlds that are live options in the current conversation).

3.2.2 Parsing

The informal description sketched above suggested that we need to reason about the truth or falsity of a sentence S at various points during its parse, even if we do not yet have access to a completed version of S. For this reason, give ourselves access to partial syntactic objects that represent the *Parse* of S. The notion of *Parse* will be built out of the auxiliary notions of *Atomic Parsing Units*, and *Decomposition*. We turn to these definitions below:

Definition 4: Atomic Parsing Unit
The *Atomic Parsing Units* are:

- The left and right parentheses: (,)
- The connectives: and, or, not
- The symbol: if
- The dot: .

We will look at a sentence S as a sequence of atomic parsing units. The length of S ($length(S)$) will be the number of atomic parsing units in S. We now define the *Decomposition* of a sentence S, as the list composed of all the atomic parsing units of a sentence in the order in which they appear in the sentence:

Definition 5: Decomposition
The *Decomposition* of a sentence S, written as $Dcmp(S)$, is a list $[\alpha_1, \dots, \alpha_n]$, $1 \leq n \leq length(S)$, such that each a_n is the n-th atomic parsing unit of S.

For instance, for $S = (p'p\ or\ q)$, $Dcmp(S)$ will be $[(,\ p'p,\ or,\ q,\ )]$. We will use the notation L_i to refer to the ith element of a list L. We now define the *Parse* of a sentence S:

Definition 6: Parse
The *Parse* of a sentence S, written as $\mathscr{P}(S)$, is a list $[\alpha_1, \dots, \alpha_n]$ such that:

- $\mathscr{P}(S)_1 = Dcmp(S)_1$
- $\mathscr{P}(S)_i = \mathscr{P}(S)_{i-1} \frown Dcmp(S)_i$, where $1 < i \leq length(S)$, and $\frown$ indicates concatenation

Thus, the *Parse* of a sentence S is the list that results by starting from the first atomic parsing unit of S, and successively concatenating to it the next parsing unit. Thus, for $S = (p'p\ or\ q)$, we get $\mathscr{P}(S) = [$`(, (p’p, (p’p or, (p’p or q, (p’p or q)`$]$.[4] The *Parsing Steps/Points* of a sentence S are elements of $\mathscr{P}(S)$. For example, the 3rd parsing point of $(p'p\ or\ q)$ is $t_3 =$ `(p’p or`.

3.2.3 Locally determined truth

For a given sentence S, at each parsing point $t_i \in \mathscr{P}(S)$, we want to be calculating the worlds (in a given context C) where S is True/False for all possible continuations (good finals); call this notion the *locally determined truth*. We will call the set of worlds where S is already True at a parsing point t_i, $\mathbb{T}$, while the set of worlds where the sentence is already False at some parsing point t_i, $\mathbb{F}$.

Definition 7: $\mathbb{T}$
The $\mathbb{T}$ set of a sentence S given a context C and Parsing Step t of S, written as $\mathbb{T}^S_t$, is the set $\{w|\ w \in C,\ and\ for\ all\ good\ finals\ d,\ td\ is\ T\ in\ w\}$.

Definition 8: $\mathbb{F}$
The $\mathbb{F}$ set of a sentence S given a context C and Parsing Step t of S, written as $\mathbb{F}^S_t$, is the set $\{w|\ w \in C,\ and\ for\ all\ good\ finals\ d,\ td\ is\ F\ in\ w\}$.

3.2.4 The presupposition constraint

The presupposition constraint that we sketched earlier requires comparing the version of S with presuppositions, to the version of S without presuppositions. Since presuppositions are the primed bits, we define an operator, $(S)^-$, that removes any primed bits of the sentence:

Definition 9: $(\mathbf{S})^-$
Given a sentence S:

- If S has the form p, $(S)^- = p$.
- If S has the form $p'p$, $(S)^- = p$.
- If S has the form $(not\ \alpha)$, $(S)^- = (not\ (\alpha)^-)$.
- If S has the form $(\alpha\ *\ \beta)$, where $* \in \{and,\ or\}$, $(S)^- = ((\alpha)^-\ *\ (\beta)^-)$.
- If S has the form $(if\ \alpha.\ \beta)$, $(S)^- = (if\ (\alpha)^-.\ (\beta)^-)$.

We will sometimes informally (and slightly abusing terminology) refer to S as the [+presup] version of a sentence, and to $(S)^-$ as the [-presup] version.

Given a presuppositions-carrying sentence $p'p$, we want to check the informativity of p' no matter the p. So we need to be reasoning about a version of the sentence where any sentence can substitute for p. To do this we define a substitution operation:

[4] We use the `verbatim` font to refer to partial syntactic objects.

Definition 10: Substitution
For every sentences S, S', for every sentence D, $S(S'|D)$ (the substitution of D for S' in S) is defined as follows:

- If S has the form p_i and $S' = p_i$, then $S(S'|D) = D$; otherwise $S(S'|D) = p_i$.
- If S has the form $p'_i p_j$ and $S' = p'_i p_j$, then $S(S'|D) = p'_i D$; otherwise $S(S'|D) = p'_i p_j$.
- If S has the form $(not\ \alpha)$, then $S(S'|D) = (not\ \alpha(S'|D))$.
- If S has the form $(\alpha\ and\ \beta)$, then $S(S'|D) = (\alpha(S'|D)\ and\ \beta(S'|D))$.
- If S has the form $(\alpha\ or\ \beta)$, then $S(S'|D) = (\alpha(S'|D)\ or\ \beta(S'|D))$.
- If S has the form $(if\ \alpha.\ \beta)$, then $S(S'|D) = (if\ \alpha(S'|D).\ \beta(S'|D))$.

We can now give a formal statement of the presupposition constraint:

Definition 11: Presupposition constraint
Given a sentence S, and $p'_i p_j$ in S, we require that for every sentence D, for every parsing point t_k in $\mathscr{P}(S(p'_i p_j|D))$, for every parsing point t'_k in $\mathscr{P}((S(p'_i p_j|D))^-)$, and for every good final d:

- $\mathbb{T}_{t_k}^{S(p'_i p_j|D)}$ $(i.e.,\ \{w|\ t_k d \text{ is } T \text{ in } w\}) \subseteq \mathbb{T}_{t'_k}^{(S(p'_i p_j|D))^-}$ $(i.e.,\ \{w|\ t'_k d \text{ is } T \text{ in } w\})$
- $\mathbb{F}_{t_k}^{S(p'_i p_j|D)}$ $(i.e,\ \{w|\ t_k d \text{ is } F \text{ in } w\}) \subseteq \mathbb{F}_{t'_k}^{(S(p'_i p_j|D))^-}$ $(i.e.,\ \{w|\ t'_k d \text{ is } F \text{ in } w\})$

3.2.5 The Update algorithm

We now bring all these ingredients together and we state the formalized version of our update algorithm:

Definition 12: Update
The update of a context C with a sentence S is defined via the following algorithm:

$Update(C,S)$:

Set $C_0 := C$

For $k \in [1,\ length(\mathscr{P}(S))]$:

Set $C_k := C_{k-1} - \mathbb{F}^S_{\mathscr{P}(S)_k}$

If there exists D such that for some $p'_i p_j$ in S:

$\mathbb{F}^{S(p'_i p_j|D)}_{\mathscr{P}(S(p'_i p_j|D))_k} \not\subseteq \mathbb{F}^{(S(p'_i p_j|D))^-}_{\mathscr{P}((S(p'_i p_j|D))^-)_k}$, or

$\mathbb{T}^{S(p'_i p_j|D)}_{\mathscr{P}(S(p'_i p_j|D))_k} \not\subseteq \mathbb{T}^{(S(p'_i p_j|D))^-}_{\mathscr{P}((S(p'_i p_j|D))^-)_k}$

then return $\emptyset$

Else continue with the loop

Return C_k

The algorithm takes a sentence S and an initial context C, and for each parsing step $\mathscr{P}(S)$, it attempts to find the worlds where the sentence is already False, and exclude them. It then asks whether for all presuppositional formulas $p'_i p_j$ in S and all sentences D: all the worlds where the [+presup] version of S (with D substituted for p_j) at parsing step t_k is already True/False are also worlds where $(S)^-$, the [-presup] version, (with D substituted for p_j) is also True/False. If the answer is 'yes', then the algorithm moves to parsing step $[k+1]$, taking as the new global context the previous global context, minus all the worlds that were excluded at the previous step. If the answer is 'no', the update fails (it returns the empty set). Thus, presupposition projection is modeled as update failure; updating here is a pragmatic process, so this failure is to be understood as infelicity, rather than falsehood.

Finally, we define the notion of *Presupposition*:

Definition 13: Presupposition
A sentence S presupposes a sentence S' iff for every C such that $Update(C,S) \neq \emptyset$, then $C \models S'$.

3.3 Results

We now turn to the derivation of some results regarding the projection behavior of the connectives. To simplify notation, we will often write $\mathbb{T}_{[+presup]}$ and $\mathbb{T}_{[-presup]}$ to indicate the the $\mathbb{T}$ that can be computed for S and $(S)^-$ respectively at some parsing point t. Similarly for the $\mathbb{F}$ sets.

Fact 1: $p'p$ presupposes p'.

Consider a context C. The parsing step where we can start reasoning about truth and falsity is $p'p$. Since a presupposition is interpreted as conjoined to the assertion, the sentence is True in $\{w \in C|\ p' = 1\ and\ p = 1\}$; False in $\{w \in C|\ p' = 0\ or\ p = 0\}$. The constraint amount to requiring that:

(8) For any sentence D:

a. $\{w \in C|\ p' = 1\ and\ D = 1\} \subseteq \{w \in C|D = 1\}$ (*i.e.*, $\mathbb{T}_{[+presup]} \subseteq \mathbb{T}_{[-presup]}$)
b. $\{w \in C|\ p' = 0\ or\ D = 0\} \subseteq \{w \in C|D = 0\}$ (*i.e.*, $\mathbb{F}_{[+presup]} \subseteq \mathbb{F}_{[-presup]}$)

The first requirement is trivial. Let's focus then on the requirement that for any sentence D: $\{w \in C|\ p' = 0\ or\ D = 0\} \subseteq \{w \in C|D = 0\}$. Since this needs to hold for all D, it needs to hold for the case where D expresses a tautology. Then the constraint becomes:

(9) $\{w \in C|\ p' = 0\} \subseteq \emptyset$

This hold just in case $\{w \in C|\ p' = 0\} = \emptyset$, which amounts to requiring that $C \models p'$.

Fact 2: $(not\ p'p)$ **presupposes** p'.

The first parsing step where we can start reasoning about truth and falsity is $t_3 =$ `(not p'p`. The only possible good-final here is a closing parenthesis. Therefore, we know that the sentence is True in $\{w \in C_3|\ p' = 0\ or\ p = 0\}$, and False in $\{w \in C_3|p' = 1\ and\ p = 1\}$. The reasoning is now the same as in **Fact 1**.

Fact 3: $(p'p\ and\ q)$ **presupposes** p'.

The first parsing step where we can start reasoning about truth and falsity is $t_3 =$ `(p'p and`. At this point we know that for any possible good final, the sentence is already False in $\{w \in C_3|\ p' = 0\ or\ p = 0\}$. Therefore, for any sentence D substituting for p, we can calculate an $\mathbb{F}$ set at parsing point $t_3 =$ `(p'D and`. The same reasoning as in **Fact 1** applies, hence $C \models p'$. This requirement arises as soon as the first conjunct plus the 'and' are parsed. It doesn't matter what the second conjunct is (hence the asymmetry in conjunction).

Fact 4: $(q\ and\ p'p)$**, where** $q \models p'$**, has trivial presuppositions.**

Again, the first step where we can start reasoning about truth and falsity is $t_3 =$ `(q and`.

We can only calculate an $\mathbb{F}$ set at this point: $\{w \in C_3|\ q = 0\}$. Per our update algorithm, we remove all these worlds from C_3, and we go on with the update taking $C_4 = C_3 - \{w|\ q = 0\}$ as the new context.

Since q carries no primed bits, there is no presuppositional expression that the constraint can target at parsing point $t_3 =$ `(q and`, so it's satisfied trivially.

We then move on to $t_4 =$ `(q and p'p`. The only possible good final here is). The [+presup] version of the sentence is True in $\{w \in C_4|q = 1\ and\ p' = 1\ and\ p = 1\}$. The [-presup] version is True $\{w \in C_4|q = 1\ and\ p = 1\}$. For any D substituting for p, it will hold that:

(10) $\{w \in C_4|q = 1\ and\ p' = 1\ and\ D = 1\} \subseteq \{w|q = 1\ and\ D = 1\}$

Thus, no issue arises for the $\mathbb{T}$ part of the constraint.

At the same parsing point, [+presup] is False $\{w \in C_4|\ q = 0\ or\ p' = 0\ or\ p = 0\}$. The [-presup] version, is False in $\{w \in C_4|\ q = 0\ or\ p = 0\}$.

But note that at this stage in the update, all the worlds where q is False have been removed, so the context only contains worlds where q is True; and since $q \models p'$, all these worlds are worlds where p' is True. Since there are no worlds where p' is False, $\{w|\ p' = 0\} = \emptyset$. Thus the sets become:

(11) a. $\mathbb{F}_{[+presup]}$: $\{w \in C_4|\ q = 1\ and\ (p' = 0\ or\ p = 0)\} = \{w|\ (q = 1\ and\ p' = 0)\ or\ (q = 1\ and p = 0)\} = \{w|\ q = 1\ and p = 0\}$ (since $\{w|\ (q = 1\ and\ p' = 0)\} = \emptyset$, as all worlds are p' worlds)

b. $\mathbb{F}_{[-presup]}$: $\{w \in C_4|\ q = 1\ and p = 0\}$

Clearly, for any D substituting for p:

(12) $\{w \in C_4 |\ q = 1\ and D = 0\} \subseteq \{w |\ q = 1\ and D = 0\}$

So, $(q\ and\ p'p)$, where $q \models p'$ can update any context C and hence it can only presuppose things that are true in all contexts (i.e., tautologies). Another way to say this is that p' gets filtered.

Fact 5: $(p'p\ or\ q)$**, where** $\neg q \models p'$**, has trivial presuppositions.**

At parsing step $t_3 =$ `(p'p or` we cannot yet compute a set of worlds where the entire sentence is False for all possible continuations. But we can compute a set where it's True:

(13) $\mathbb{T}_{[+presup]} = \{w \in C_3 |\ p' = 1\ and\ p = 1\}$.

(14) $\mathbb{T}_{[-presup]} = \{w \in C_3 |\ p = 1\}$.

For any D:

(15) $\{w \in C_3 |\ p' = 1\ and\ D = 1\} \subseteq \{w \in C_3 |\ D = 1\}$

We move to $t_4 =$ `(p'p or q`. The only good final here is). Thus:

(16) $\mathbb{T}_{[+presup]} = \{w \in C_4 |\ (p' = 1\ and\ p = 1)\ or\ q = 1\} = \{w \in C_4 |\ (p' = 1\ or\ q = 1)\ and\ (p = 1\ or\ q = 1)\}$.

(17) $\mathbb{T}_{[-presup]}\{w \in C_4 |\ p = 1\ or\ q = 1\}$.

Again for all D:

(18) $\{w \in C_4 |\ (p' = 1\ or\ q = 1)\ and\ (D = 1\ or\ q = 1)\} \subseteq \{w \in C_4 |\ D = 1\ or\ q = 1\}$.

Things are more interesting when we consider the $\mathbb{F}$ set. For the [+presup] version it is:

(19) $\mathbb{F}_{[+presup]} = \{w \in C_4 |\ (p' = 0\ or\ p = 0)\ and\ q = 0\} = \{w \in C_4 | (p' = 0\ and\ q = 0)\ or\ (p = 0\ and\ q = 0)\}$.

(20) $\mathbb{F}_{[-presup]} = \{w \in C_4 |\ p = 0\ and\ q = 0\}$.

Since we assumed that $\neg q \models p'$, we cannot have worlds where q is False and p' is False; hence, $\{w \in C_4 | p' = 0\ and\ q = 0\} = \emptyset$. Thus, $\mathbb{F}_{[+presup]} = \{w \in C_4 |\ p = 0\ and\ q = 0)\}$. Hence, for all D:

(21) $\{w \in C_4 | D = 0 \textit{ and } q = 0\} \subseteq \{w \in C_4 | D = 0 \textit{ and } q = 0\}$

A similar argument establishes this result for the $(q \textit{ or } p'p)$ case, where again $\neg q \models p'$.

Fact 6: $(\textit{if } p'p.\ q)$ **presupposes** p'**.**

At parsing step $t_4 =$ `(if p'p.` we can calculate a $\mathbb{T}$ set for [+presup], namely $\{w\,|p'(w) = 0 \textit{ or } p(w) = 0\}$, since in all worlds where the antecedent fails, the entire conditional is automatically True. The corresponding set for [-presup] is $\{w \mid p(w) = 0\}$.

Our constraint requires, for all D:

(22) $\{w|p' = 0 \textit{ or } D = 0\} \subseteq \{w|\ D = 0\}$.

Again, the reasoning here is the same as in **Fact 1**.

Fact 7: $(\textit{if } (\textit{not } p'p).\ q)$**, where** $\neg q \models p'$**, has trivial presuppositions.**

At parsing step $t_6 =$ `(if (not p'p.`, we can calculate a $\mathbb{T}_{[+presup]} = \{w \in C_6|\ p' = 1 \textit{ and } p = 1\}$; for the [-presup] version, this is $\{w \in C_6 | p = 1\}$. For all D then:

(23) $\{w \in C_6|\ p' = 1 \textit{ and } D = 1\} \subseteq \{w \in C_6 | D = 1\}$

At parsing step $t_7 =$ `(if (not p'p. q`, we can calculate:

(24) $\mathbb{T}_{[+presup]} = \{w \in C_7|\ (p' = 1 \textit{ and } p = 1) \textit{ or } q = 1\} = \{w \in C_7|\ (p' = 1 \textit{ or } q = 1) \textit{ and } (p = 1 \textit{ or } q = 1)\}$.

(25) $\mathbb{T}_{[-presup]} = \{w \in C_7|\ p = 1 \textit{ or } q = 1\}$.

Again, for all D substituting for p, the direction of subsethood is the correct one.

Consider now the $\mathbb{F}$ sets:

(26) $\mathbb{F}_{[+presup]} = \{w \in C_7|\ (p' = 0 \textit{ or } p = 0) \textit{ and } q = 0\} = \{w \in C_7|\ (p' = 0 \textit{ and } q = 0) \textit{ or } (p = 0 \textit{ and } q = 0\}$.

(27) $\mathbb{F}_{[-presup]} = \{w \in C_7|\ p = 0 \textit{ and } q = 0\}$.

Since, $\neg q \models p'$, $\{w \in C | p' = 0 \textit{ and } q = 0\} = \emptyset$. Therefore, the two sets are contextually equivalent for all D substituting for p (just like the disjunction case), and no projection arises.

Note that such cases of essentially 'bathroom conditionals' exist (Schlenker 2009):

(28) If the bathroom is not hidden, then there is no bathroom. (no projection of the presupposition in the antecedent that 'there is a bathroom')

Thus our system derives that presuppositions project from negation, the first conjunct, and the antecedent of simple conditionals, while at the same time the mechanism allows enough flexibility to avoid projection in cases of ‘bathroom disjunctions/conditionals’.

4. Conclusion

We have formalized a *bivalent system* for presupposition projection that is *parsing-oriented*, and captures the basic projection behavior of the connectives in a way that makes *conjunction asymmetric, but disjunction symmetric through a single mechanism*. The system can also capture cases of symmetric conditionals. Of courser, the system’s predictions need to be investigated further (testing it for example on antecedent-final conditionals,(Mandelkern and Romoli 2017), and extending it to quantification), to get a clearer view of its empirical ‘bite’. Nevertheless, its successes with the simple cases presented in this paper, are clear.

References

Heim, Irene. 1983. On the projection problem for presuppositions. In *Proceedings of WCCFL 2*, ed. by M. Barlow, D. Flickinger, and N. Wiegand, 114–125. Stanford, CA.

Kalomoiros, Alexandros, and Florian Schwarz. 2021. Presupposition projection from disjunction is symmetric. In *Proceedings of the Linguistic Society of America*, volume 6, 556–571.

Karttunen, Lauri. 1973. Presuppositions of compound sentences. *Linguistic Inquiry* 4:169–193.

Karttunen, Lauri. 1974. Presupposition and linguistic context. *Theoretical Linguistics* 1:181–194.

Langendoen, D. Terence, and Harris Savin. 1971. The projection problem for presuppositions. In *Studies in linguistic semantics*, ed. by Charles J. Fillmore and D. Terence Langėndoen, 54–60. Irvington.

Mandelkern, Matt, and Jacopo Romoli. 2017. Parsing and presuppositions in the calculation of local contexts. *Semantics and Pragmatics* 10.

Mandelkern, Matthew, Jérémy Zehr, Jacopo Romoli, and Florian Schwarz. 2020. We’ve discovered that projection across conjunction is asymmetric (and it is!). *Linguistics and Philosophy* 43:473–514.

Rothschild, Daniel. 2011. Explaining presupposition projection with dynamic semantics. *Semantics and Pragmatics* 4:1–43.

Schlenker, Philippe. 2008. Be articulate: A pragmatic theory of presupposition projection. *Theoretical Linguistics* 34:157–212.

Schlenker, Philippe. 2009. Local contexts. *Semantics and Pragmatics* 2:1–78.

Stalnaker, Robert. 1974. Pragmatic presuppositions. In *Semantics and Philosophy*, ed. by Milton K. Munitz and Peter K. Unger, 197–213. New York University Press.

Alexandros Kalomoiros
akalom@sas.upenn.edu