Limited Symmetry*

Alexandros Kalomoiros[0000−0002−3861−8514]

University of Pennsylvania, Philadelphia PA 19104, USA
akalom@sas.upenn.edu

Abstract. This paper aims to tackle some of the basic (a-)symmetries of presupposition projection in a pragmatic, bivalent, and incrementally-oriented framework. The main data point that we are trying to capture is that projection from the first conjunct of a conjunction is asymmetric, while projection from the first disjunct of a disjunction can be symmetric. We argue that a solution where there are effectively two filtering mechanisms, one symmetric and one asymmetric, [10], is not tenable given recent experimental evidence, [6, 3]. Instead, we propose a bivalent system, where at each point during the incremental interpretation of a sentence S, the comprehender is trying to compute the sets of worlds in the context where the truth value of S has already been determined. This computation plays out differently in the case of conjunction vs the case of disjunction, and coupled with appropriate definitions of the incremental interpretation process and of what it means for a presupposition to project in the current system, it leads to asymmetric conjunction, but symmetric disjunction.

Keywords: Presupposition Projection · (A-)symmetries · Connectives.

1 Introduction

This paper is concerned with the projection problem for presuppositions. Certain lexical items impose conditions (presuppositions) on the context in which they are uttered, (see e.g. [12] among many others):

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

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

* Thanks to Florian Schwarz for many helpful discussions and comments on this work. Thanks also to Phillipe Schlenker, Jacopo Romoli, Julie Legate, Anna Papafragou, Ryan Budnick, Andrea Beltrama, Spencer Caplan, and to the members of the Penn Semantics Lab for helpful comments and suggestions. Thanks also to the ESSLLI reviewers for useful feedback and suggestions. All errors are my own.

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?

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

Example (3-b) 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 (3-b) 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 (3-c) is not felt to carry the presupposition that John used to have research interests in Tolkien. In this case we say that the presupposition is **filtered** (in the terminology of Karttunen, [4]). The only difference between (3-b) and (3-c) is the order of the conjuncts: in (3-b) the presupposition-bearing conjunct comes first (see also [6] on experiments that confirm this difference for conjunction).

Thus, conjunction represents a case where the filtering of presuppositions is asymmetric. An intuitive explanation for this is that the the first conjunct in (3-c) is evaluated first ([12]), 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 (3-b) 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, [4]:

(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.

Neither (4) nor (5) are felt to presuppose that John used to have research interests in Tolkien. This raises the following conundrum: projection from conjunction would appear to imply that the mechanism determining projection/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 is then is whether need two different filtering mechanisms, one symmetric and one asymmetric, to account for this landscape (as proposed by [10]); 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 [10]'s theory of local contexts, which aims to tackle the problem in a pragmatic, incrementally-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 (dubbed 'Limited Symmetry') within the a pragmatic, incrementally-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 Schlenker's approach to the projection problem [10], as our own solution in section 3 follows the spirit of this approach. Schlenker provides a reconstruction of Karttunen's notion of 'local context', [4]. **The 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 will, of course, depend on the truth-functional properties of the connective in question, which ultimately accounts 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.

[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.

Schlenker assumes a 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, [13]. 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', $C \models a(r \textit{ and } D)b' \leftrightarrow a(D)b'$

Now consider a conjunction like $(p \textit{ and } q)$, 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', $C \models (p \textit{ and } (r \textit{ and } D)b' \leftrightarrow (p \textit{ and } (D)b'$. The only possible good final here is a closing parenthesis,). One proposition that does the job is the proposition expressed by p: $C \models (p \textit{ and } (p \textit{ and } D)b' \leftrightarrow (p \textit{ and } (D)b'$, since conjoining the first conjunct to the second conjunct is not going to change the truth conditions of the conjunction. 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, $(p \textit{ and } D)$ is true in w', but $(p \textit{ and } (r \textit{ and } D))$ is false; so it does not hold that $C \models (p \textit{ and } D) \leftrightarrow (p \textit{ and } (r \textit{ and } D))$. Therefore, the local context for a second conjunct is the first conjunct.

Applying similar reasoning, we can calculate the local context for the second disjunct q of a disjunction $(p \textit{ or } q)$: $C \models (p \textit{ or } (D)b' \leftrightarrow (p \textit{ or } ((\textit{not } p) \textit{ and } D)b'$.[3] $(\textit{not } p)$ is the strongest proposition we could conjoin here: suppose we also conjoin r such that there is a C-world w, where r is false, p is false, but D is true (again the only possible good-final here is the closing parenthesis). In this world, $(p \textit{ or } ((\textit{not } p) \textit{ and } D)$ is true but $(p \textit{ or } ((\textit{not } p) \textit{ and } r \textit{ and } D)$ is not. So, the local context of a second disjunct is the negation of the first disjunct, thus correctly predicting filtering in cases like (5).

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 $C \models ((D)b' \leftrightarrow ((r \textit{ and } D)b'$, for all possible good finals 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 we get into

[2] This definition focuses on the propositional case and is borrowed from [5]. See [10] for full definitions generalized to a more expressive language.

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

trouble: consider a proposition r which is false in some C-world w, while D is true in w and $b' = and\ \top$, where $\top$ is a tautology. Then $(r\ and\ D\ and\ \top)$ is false in w, but $(D\ and\ \top)$ is true in w, so the equivalence between them fails. Thus, a presupposition in the first conjunct/disjunct creates a problem unless it is 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, [10] 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, [10] posits that asymmetric local contexts are the default (it is in this sense that the system is parsing-oriented), while accessing symmetric contexts involves overriding this default and hence carries a processing cost.

Nevertheless, recent experimental evidence casts doubt on this claim. Recent work by Mandelkern et al, [6], has shown 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. [3] 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. Therefore, it seems preferable to derive the projection (a-)symmetries via a single mechanism.

3 Limited Symmetry

3.1 The system informally

We develop a pragmatic, bivalent, incrementally-oriented system, which we dub ‘Limited Symmetry’, inspired by Schlenker’s approach, 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, 3.3.

Part of the reason that the asymmetric definition of local contexts could not differentiate between the first conjunct vs the first disjunct was that one does does not have access to the connective when calculating those local contexts, as the connective comes after the first conjunct/disjunct. To overcome this, 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).[4]

To do this we will assume that when encountering a sentence, e.g. a conjunction like $S = (p\ and\ q)$, comprehenders parse the symbols of S incrementally: [`(`, `(p`, `(p and`, `(p and q`, `(p and q)`]. Similarly, a disjunction of the form $(p\ or\ q)$ is parsed as [`(`, `(p`, `(p or`, `(p or q`, `(p or q)`].[5] Each entry in these lists represents a parsing step for the given sentence. Let's assume that at each parsing step the comprehender 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.

Our treatment of presuppositions is very much related to Schlenker's. Recall that a (symmetric) local context is the strongest proposition one can conjoin to a sub-constituent D without changing the overall truth conditions of the sentence S in which D is embedded (for any D). One can add or remove a local context to its sub-constituent *salva veritate*, no matter the identity of the sub-constituent. Presuppositions of D then must be entailed by the local context. Equivalently, a local context is the strongest proposition that D can presuppose; presuppositions then are essentially subject to a constraint that says that they can be removed without changing the truth conditions of S.[6] [7]

We will assume that some sentences carry presuppositions, represented as $p'p$, where p' is the presuppositional component of the sentence, and p the assertive

[4] Throughout the rest of this paper, the `verbatim` font is used to mark partial syntactic objects.

[5] Note that I sometimes use the term 'parsing' to mean roughly 'getting access to bits of syntactic structure during incremental interpretation', which deviates from the common usage of the term somewhat. Nonetheless, within the presuppositions literature, approaches like that of [10] that work by manipulating partial strings of expressions, are often referred to as 'parsing-based' (see e.g. [5]), and it is this usage I have in mind here.

[6] Schlenker himself originally stated his constraint on presuppositions in such terms, in the context of his Transparency theory, [8, 9]. The local contexts theory is an equivalent reformulation of the Transparency theory, as shown in Schlenker 2009.

[7] There are similarities here with the Strong Kleene algorithm, [2], where projection (i.e., the third truth value) results when the other sentences of a larger sentence are not enough for determining the classical truth value, given the semantics of the connective. In fact, as discussed for instance in [7], the predictions of Strong Kleene and symmetric Local Contexts are very close to one another.

component. Following [10], we will understand such sentences as the conjunction of p' and p in a bivalent classical logic. Consider the effect of the constraint we described in the previous paragraph on $p'p$; it says that for any p,[8] p' can be added or removed from p without change in truth conditions:

(6) $\forall p : C \models p'p \leftrightarrow p$

We can re-write this in the following way:

(7) For all p:

a. $\{w \in C|\ p' = 1\ \mathit{and}\ p = 1\} \subseteq \{w \in C|\ p = 1\}$ (this is simply $C \models p'p \rightarrow p$)

b. $\{w \in C|\ p' = 0\ \mathit{or}\ p = 0\} \subseteq \{w \in C|\ p = 0\}$ (this is the contrapositive of $C \models p \rightarrow p'p$)

So, this Schlenkerian constraint boils down to saying that for all p, all of the worlds in the context where the sentence with the presupposition p' is true should be worlds where the sentence without p' is true, and similarly for false. It can be shown (see Observation 1 in 3.3) that this constraint is satisfied just in case $C \models p'$, i.e. just in case the context entails the presuppositions of the sentence, which is the desired result.

Here's now our twist on this idea: the aim of the comprehender is indeed to show that for every presuppositional component $p'p$ of a sentence S, for all p the version of S with the presupposition p' (the [+presup] version), is equivalent to the version without p' (the [-presup] version). However, instead of waiting until they have access to the whole of S before attempting to check this equivalence, they try to build it incrementally as they are getting access to the worlds where S is true or false.

For example, suppose that $S = (p'p\ \mathit{and}\ q)$. The general requirement is that in a context C, $(p'p\ \mathit{and}\ q)$ should be equivalent to $(p\ \mathit{and}\ q)$, for all p. So, all worlds in C where $(p'p\ \mathit{and}\ q)$ is true should be worlds where $(p\ \mathit{and}\ q)$ is true (for all p); and similarly all worlds where $(p'p\ \mathit{and}\ q)$ is false should be worlds where $(p\ \mathit{and}\ q)$ is false (for all p).

The key is that a comprehender doe not need access to the whole of S to start calculating whether this equivalence holds. At the point in parse where the comprehender has access only to the `(p'p and` part of S they already know that the sentence is false in all C-worlds where $p'p$ is false. Therefore, for these worlds the comprehender can already at this point check whether they satisfy the condition that they should also be worlds where the [-presup] version of S is false. The [-presup] version that the comprehender has access to at this point is `(p and`. The question then is whether for all p: $\{w|p' = 0\ \mathit{or}\ p = 0\}$ is (contextually) a subset of $\{w|p = 0\}$. It can be shown (see section 3.3) that this happens just in case $C \models p'$, which is the desired result.

Conversely, given a disjunction $S = (p'p\ \mathit{or}\ q)$, at parsing step `(p'p or`, we can determine a set of worlds where the disjunction is true for all possible

[8] This is just the 'for any D' part in the definitions of local contexts

continuations: all the worlds where p' is true and p is true: $\{w| \; p' = 1 \; and \; p = 1\}$, which for all p, is a subset of $\{w| \; p = 1\}$ (the corresponding set if we only consider the [-presup] version of S at the corresponding parsing point). This already shows that our system distinguishes between conjunction and disjunction: they have different points where the conditions imposed on presuppositions can create trouble (see section 3.3 for how exactly symmetry comes about). We now turn to formalizing 'Limited Symmetry'.

3.2 Formalization

We restrict ourselves to a propositional language $\mathcal{L}$ (inspired by [10]):

(8) a. **Propositions** $:= p_i \mid p'_j p_k$ (subscripts are natural numbers)
  b. **Formulas** $\phi := \; (not \; \phi) \mid (\phi \; and \; \phi) \mid (\phi \; or \; \phi) \mid (if \; \phi. \; \phi)$

In $p'_j p_k$, p'_i is meant to be understood as the an entailment that is marked as a presupposition (hence the prime), while p_k as the non-presuppositonal entailment (the asserted content). 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 $\langle W, F \rangle$, where W is a set of worlds, and F is a function assigning to each propositional constant of $\mathcal{L}$ a set of worlds. Our semantics is bivalent and follows the standard truth tables. Presupposition-bearing sentences are treated as conjunctions:

Definition 3: Satisfaction

- $w \models p$ iff $w \in F(p)$
- $w \models p'p$ iff $w \in F(p')$ and $w \in F(p)$
- $w \models (not \; \phi)$ iff $w \not\models \phi$
- $w \models (\phi \; and \; \psi)$ iff $w \models \phi$ and $w \models \psi$
- $w \models (\phi \; or \; \psi)$ iff $w \models \phi$ or $w \models \psi$
- $w \models (if \; \phi. \; \psi)$ iff $w \not\models \phi$ or $w \models \psi$

We follow [10] 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). Recall that we want access to the *parsing steps* of a sentence S, which we represent as a sequence of substrings of S. We define this list in three steps. First we define the notion of an *atomic parsing unit*:

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 then define the $Parse$ of a sentence S:

Definition 5: Parse The Parse of a sentence S, written as $P(S)$, is a list $[\alpha_1, \ldots, \alpha_{length(S)}]$, such that each a_i is the ith atomic parsing unit of S.

For instance, for $S = (p'p \text{ } or \text{ } q)$, $P(S)$ 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 parsing steps of a sentence S:

Definition 6: Parsing Steps The Parsing Steps of a sentence S, written as $PS(S)$, are a list $[\alpha_1, \ldots, \alpha_n]$ such that:

- $PS(S)[1] = P(S)[1]$
- $PS(S)[i] = PS(S)[i-1]^\frown P(S)[i]$, where $1 < i \leq length(S)$, and $^\frown$ indicates concatenation

Thus, the parsing steps 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 $S = (p'p \text{ } or \text{ } q)$, $PS(S) =$ [`(`, `(p`, `(p or`, `(p or q`, `(p or q)`].

At each parsing step of S, we want to be calculating the worlds (in a given context C) where S is true/false for all possible continuations (good finals). We will call the set of worlds where S is already true, the *Includable Context*, while the set of worlds where the sentence is already false the *Excludable Context*.

Definition 7: Includable Context The Includable Context given a context C and Parsing Step t, written as $IC(C, t)$, is the set $\{w| \text{ } w \in C, \text{ } and \text{ } for \text{ } all \text{ } good \text{ } finals \text{ } d, \text{ } w \models td\}$.

Definition 8: Excludable Context The Excludable Context given a context C and Parsing Step t, written as $EC(C, t)$, is the set $\{w| \text{ } w \in C, \text{ } and \text{ } for \text{ } all \text{ } good \text{ } finals \text{ } d, \text{ } w \not\models td\}$.

The constraint we described earlier depends on comparing the version of the sentence with the presupposition to the version of the sentence where presuppositional sentences have been replaced by their non-presuppositional counterparts. It will be useful to stipulate that every instance of a presuppositional constant in a sentence S is unique.[9] We define a substitution operation, $S_{p'p/p}$, that substitutes an atomic sentence of the form $p'p$ in S with p. Since sentences of the $p'p$ form are unique in S, every substitution has only one instance:[10]

[9] This leads to no loss of generality. Every time a sentence contains the same $p'_i p_j$ symbol in two different positions, just rewrite S with one of the $p'_i p_j$ instance changed to $p_k p_j$, where $i \neq k$, with the stipulation that F assigns to both p_i and p_k the same set of worlds.

[10] A reviewer points out that this way of thinking about presuppositions (as *representationally distinct* from assertions) is not necessarily compatible with triggering algorithms (e.g., [1, 11]), which view presuppositions as entailments of the overall proposition that are selected by the triggering algorithm and marked as presuppositions. Nevertheless, I think one could view the process I'm describing here as what happens *once an entailment has been triggered as a presupposition*, essentially taking triggering for granted and focusing on a representation where what is to be presupposed has been already marked; in effect this follows [10] in separating conceptually the triggering problem from the projection problem.

Definition 9: $S_{p'p/p}$ Given a sentence S and an atomic proposition $p'p$:

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

Finally, we can state our update algorithm more precisely:

Definition 10: 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 $i \in [1, length(PS(S))]$:
    For every $p'p$ in $PS(S)[i]$:
    If $\forall p: EC(C_{i-1}, PS(S)[i]) \nsubseteq EC(C_{i-1}, PS(S_{p'p/p})[i])$
    or $IC(C_{i-1}, PS(S)[i]) \nsubseteq IC(C_{i-1}, PS(S_{p'p/p})[i])$
    then return #
    Else continue with the loop
    Set $C_i := C_{i-1} - EC(C_{i-1}, PS(S)[i])$
  Return C_i

The algorithm takes a sentence S and an initial context C, and for each parsing step $PS(S)[i]$, it attempts to find the worlds where the sentence is already false, and exclude them. It doing this, it checks whether for all $p'p$ in S up to that point in the parse, for all p:

(9) All the worlds that are excluded/included by the [+presup] version of S at parsing step $PS(S)[i]$ are also excluded/included by the [-presup] version of S, namely $S_{p'p/p}$, at parsing step $PS(S_{p'p/p})[i]$.

If yes, then the algorithm moves to parsing step $[i+1]$, taking as the new global context the previous global context, minus all the worlds that were excluded at the previous step. If no, the update fails (it returns an undefinedness value #). Thus, presupposition projection is modeled as update failure; updating here is a pragmatic process, so this failure is to be understood as infelicity.

3.3 Results

We now turn to the derivation of some results regarding the projection behavior of the connectives.

Observation 1: Given a context C and $\mathbf{S = p'p}$, $\mathbf{Update(C, S) \neq \#}$ just in case $\mathbf{C \models p'}$.

For the update process to return # the constraint on presuppositions built into the updating process must fail. The parsing step where we can start reasoning about Includable and Excludable contexts is $p'p$. Focus on the EC. Since a presupposition is interpreted as conjoined to the assertion, the EC for the [+presup] version is $\{w|\ p' = 0\ \mathit{or}\ p = 0\}$. The EC for the $S_{p'p/p} = p$ is $\{w|\ p = 0\}$. Our constraint checks whether:

(10) $\forall p : \{w|\ p' = 0\ \mathit{or}\ p = 0\} \subseteq \{w|\ p = 0\}$

Suppose this holds. Then it holds for all p, e.g. for $p = \top$ (where $\top$ is a tautology). The constraint then becomes:

(11) $\{w|\ p' = 0\} \subseteq \emptyset$

For (11) to hold, the context needs to contain no p' worlds, i.e. $C \models p'$. Now suppose that $C \models p'$. In this case the EC for the [+presup] version is $\{w|\ p = 0\}$. It is easy to see then that the condition in (10) holds. For the IC, it clearly holds for all p that $\{w|\ p' = 1\ \mathit{and}\ p = 1\} \subseteq \{w|\ p = 1\}$

Observation 2: Given a context C and S = (not p′p), Update(C, S) ≠ # just in case C ⊨ p′.

The reasoning is exactly parallel to **Observation 1** only now the set of worlds $\{w|\ p' = 0\ \mathit{or}\ p = 0\}$ is the IC instead of the EC.

Observation 3: Given a context C and S = (p′p and q), Update(C, S) ≠ # just in case C ⊨ p′.

The first step where we can start reasoning about IC and EC is `(p'p and`. At this point we can calculate an EC: for the [+presup] version it is $\{w|\ p' = 0\ \mathit{or}\ p = 0\}$. The [-presup] version, $S_{p'p/p}$ at the corresponding point is `(p and`. so it's EC is $\{w|\ p = 0\}$. The then constraint demands that:

(12) $\forall p : \{w|\ p' = 0\ \mathit{or}\ p = 0\} \subseteq \{w|\ p = 0\}$

At this point, the reasoning follows **Observation 1**.

Note that for both S and $S_{p'p/p}$, at this parsing point the ICs are the empty set (there are no worlds in the context where the sentence is true regardless of continuation yet), so the ICs here are equal.

Observation 4: Given a context C and S = (q and p′p), Update(C, S) ≠ # just in case C ⊨ q → p′.

Again, the first step where we can start reasoning about IC and EC is `(q and`. Since q carries no primed bits, the EC and the IC for the [+presup] version and the [-presup] will be equal, since the [+presup] version is the same as the [-presup] version.

Note that we can calculate an EC at this point: $\{w|\ q = 0\}$. Per our update algorithm, we remove all these worlds from C, and we go on with the update taking $C' = C - \{w|\ q = 0\}$ as the new context.

We then move on to `(q and p'p`. The only possible good final here is `)`. The IC for the [+presup] version is $\{w|q = 1\ \mathit{and}\ p' = 1\ \mathit{and}\ p = 1\}$. The IC for the [-presup] is $\{w|q = 1 \textit{ and } p = 1\}$. Since clearly for all p: $\{w|q = 1 \textit{ and } p' = 1 \textit{ and } p = 1\} \subseteq \{w|q = 1 \textit{ and } p = 1\}$, no issue arises.

The EC at the same parsing step for [+presup] is $\{w|\ q = 0 \textit{ or } p' = 0 \textit{ or } p = 0\}$. For the [-presup] version, it is $\{w|\ q = 0 \textit{ or } p = 0\}$. 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; therefore, the constraint demands that:

(13) $\forall p$: $\{w|\ (q = 1 \textit{ and } p' = 0) \textit{ or } (q = 1 \textit{ and } p = 0)\} \subseteq \{w|\ q = 1 \textit{ and } p = 0\}$

Assume that this holds. Then it holds for the case where $p = \top$. So, we have:

(14) $\forall p$: $\{w|\ q = 1 \textit{ and } p' = 0\} \subseteq \emptyset$

This holds just in case there are no worlds in C where $q = 1$ and $p' = 0$, or equivalently, iff $C \models q \rightarrow p'$. It is easy to check that if $C \models q \rightarrow p'$, then the condition in (13) holds.

Observation 5: Given a context C and S $= (\mathbf{p'p\ or\ q})$**, Update**$(\mathbf{C}, \mathbf{S}) \neq \#$ **just in case** $\mathbf{C} \models \neg\mathbf{q} \rightarrow \mathbf{p'}$**.**

At parsing step `(p'p or` we cannot yet compute a a set of worlds where the entire sentence is false for all possible continuations. But we can compute an IC. For the [+presup] version, $IC = \{w|\ p' = 1 \textit{ and } p = 1\}$. The corresponding set for the [-presup] version is $\{w|\ p = 1\}$. Since clearly $\forall p:\ \{w|\ p' = 1 \textit{ and } p = 1\} \subseteq \{w|\ p = 1\}$, no issue arises.

We move to `(p'p or q`. The only good final here is `)`. The IC for the [+presup] version is $\{w|\ (p' = 1 \textit{ and } p = 1) \textit{ or } q = 1\}$. For the [-presup] version, it is $\{w|\ p = 1 \textit{ or } q = 1\}$. Since clearly $\forall p:\ \{w|\ (p' = 1 \textit{ and } p = 1) \textit{ or } q = 1\} \subseteq \{w|\ p = 1 \textit{ or } q = 1\}$, no issue arises.

Things are more interesting when we consider the EC. For the [+presup] version it is $\{w|\ (p' = 0 \textit{ or } p = 0) \textit{ and } q = 0\} = \{w|(p' = 0 \textit{ and } q = 0) \textit{ or } (p = 0 \textit{ and } q = 0)\}$. For the [-presup] version, it is $\{w|\ p = 0 \textit{ and } q = 0\}$. The constraint demands that:

(15) $\forall p : \{w|(p' = 0 \textit{ and } q = 0) \textit{ or } (p = 0 \textit{ and } q = 0)\} \subseteq \{w|\ p = 0 \textit{ and } q = 0\}$

Assume that this holds. Then it needs to hold for the case where $p = \top$. Then the condition becomes:

(16) $\{w|p' = 0 \textit{ and } q = 0\} \subseteq \emptyset$

This holds iff there are no worlds in C where $q = 0$ and $p' = 0$, or in other words $C \models q \vee p'$ which is equivalent to $C \models \neg q \rightarrow p'$. Similarly, it is easy to establish that if $C \models \neg q \rightarrow p'$, then the condition in (15) holds.

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

Observation 6: Given a context C and S = (if p′p. q), Update(C, S) ≠ # just in case C ⊨ p′.

At parsing step `(if p'p.` we can calculate an IC, for [+presup], namely $\{w|p' = 0 \textit{ or } p = 0\}$, since in all worlds where the antecedent fails, the entire conditional is automatically true. The corresponding set for [-presup] is $\{w|\ p = 0\}$. The constraint then demands:

(17) $\forall p : \{w|p' = 0 \textit{ or } p = 0\} \subseteq \{w|\ p = 0\}$

Again, the reasoning here is the same as in Observation 1.

Observation 7: Given a context C and S = (if (not p′p). q), Update(C, S) ≠ # just in case C ⊨ ¬q → p′.

At parsing step `(if (not p'p`, we can calculate an IC: for the [+presup] version, this will be $\{w|\ p' = 1 \textit{ and } p = 1\}$; for the [-presup] version, this is $\{w|p = 1\}$. Since subsethood is in the correct direction, no issue arises.

At parsing step `(if (not p'p. q`, we can calculate both an IC and an EC. The [+presup] IC is $\{w|\ (p' = 1 \textit{ and } p = 1) \textit{ or } q = 1\}$. The corresponding [-presup] set is $\{w|\ p = 1 \textit{ or } q = 1\}$. The direction of subsethood in this case then is the correct one.

Consider now the EC. For [+presup], it is $\{w|\ (p' = 0 \textit{ or } p = 0) \textit{ and } q = 0\} = \{w|\ (p' = 0 \textit{ and } q = 0) \textit{ or } (p = 0 \textit{ and } q = 0\}$. For the [-presup] version, it is $\{w|\ p = 0 \textit{ and } q = 0\}$. At this point, the reasoning follows the disjunction case in **Observation 5** exactly.

Note that such cases of non-projection where the negation of q entails the presupposition p' exist (essentially 'bathroom conditionals'; see also [10]):

(18) 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 correct results for the projection behavior of presuppositions embedded under the connectives: presuppositions project from negation, the first conjunct, and the antecedent of conditionals, while at the same time the mechanism allows enough flexibility to avoid projection in cases of 'bathroom disjunctions/conditionals'.

4 Conclusion

We have developed a **bivalent system** for presupposition projection that is **incrementally-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. However, it is limited to the propositional case. A question to pursue in the future is how to generalize these intuitions to handle projection from the scope of quantifiers. Moreover, it's important to apply the system to more complex cases beyond conjunction and disjunction (e.g. antecedent-final conditionals (see [5])), and experimentally test the system's predictions in these cases, as this will afford a clearer view of the system's empirical 'bite'.

References

1. Abrusán, M.: Predicting the presuppositions of soft triggers. Linguistics and Philosophy **34**(6), 491–535 (2011)
2. George, B.: Presupposition repairs: a static, trivalent approach to predicting projection. UCLA MA thesis (2008)
3. Kalomoiros, A., Schwarz, F.: Presupposition projection from disjunction is symmetric. Proceedings of the Linguistic Society of America **6**(1), 556–571 (2021)
4. Karttunen, L.: Presuppositions of compound sentences. Linguistic inquiry **4**(2), 169–193 (1973)
5. Mandelkern, M., Romoli, J.: Parsing and presuppositions in the calculation of local contexts. Semantics and Pragmatics **10**, 7 (2017)
6. Mandelkern, M., Zehr, J., Romoli, J., Schwarz, F.: We've discovered that projection across conjunction is asymmetric (and it is!). Linguistics and Philosophy **43**(5), 473–514 (2020)
7. Rothschild, D.: Presupposition projection and logical equivalence. Philosophical Perspectives **22**, 473–497 (2008)
8. Schlenker, P.: Anti-dynamics: Presupposition projection without dynamic semantics. Journal of logic, language and information **16**, 325–356 (2007)
9. Schlenker, P.: Be articulate: A pragmatic theory of presupposition projection. Theoretical Linguistics pp. 157–212 (2008)
10. Schlenker, P.: Local contexts. Semantics and Pragmatics **2**, 3–1 (2009)
11. Schlenker, P.: Triggering presuppositions. Glossa: a journal of general linguistics **6**(1) (2021)
12. Stalnaker, R.: Pragmatic presuppositions. In: Munitz, M.K., Unger, P.K. (eds.) Semantics and Philosophy, pp. 197–213. New York University Press (1974)
13. Stalnaker, R.: Assertion. Syntax and Semantics (New York Academic Press) **9**, 315–332 (1978)