Finite State Reasoning for Presupposition Satisfaction

Jacob Collard
Cornell University
jnc76@cornell.edu

Abstract

Sentences with presuppositions are often treated as uninterpretable or unvalued (neither true nor false) if their presuppositions are not satisfied. However, there is an open question as to how this satisfaction is calculated. In some cases, determining whether a presupposition is satisfied is not a trivial task (or even a decidable one), yet native speakers are able to quickly and confidently identify instances of presupposition failure. I propose that this can be accounted for with a form of possible world semantics that encapsulates some reasoning abilities, but is limited in its computational power, thus circumventing the need to solve computationally difficult problems. This can be modeled using a variant of the framework of finite state semantics proposed by Rooth (2017). A few modifications to this system are necessary, including its extension into a three-valued logic to account for presupposition. Within this framework, the logic necessary to calculate presupposition satisfaction is readily available, but there is no risk of needing exceptional computational power. This correctly predicts that certain presuppositions will not be calculated intuitively, while others can be easily evaluated.

1 Introduction

Accounts of presupposition are typically concerned with describing the contexts in which a presupposition is satisfied, and with the syntactic and compositional factors which relate to the projection properties of presuppositions. However, there are a number of issues that can arise using the highly general methods for calculating presupposition satisfaction preferred by these accounts. Though many previous accounts roughly outline the sets in which a presupposition may be satisfied, they are not restrictive enough to allow for an actual computational implementation or to explain the cognitive reality of presupposition satisfaction.

Early work characterized presuppositions as relations between sentences and logical forms where a sentence A and a logical form L would be related iff A could only be uttered in contexts where L was entailed (Karttunen, 1973). Karttunen suggested a notion of presupposition satisfaction based on entailment, claiming that a context would satisfy the presuppositions of a sentence just in case the context entailed all of the basic presuppositions of the sentence. However, Karttunen does not explicitly define how the logical forms entailed by a context are calculated. Instead, he simply defines the context as "a set of logical forms that describe the set of background assumptions, that is, whatever the speaker chooses to regard as being shared by him and his intended audience." How a speaker determines this set of logical forms notwithstanding, it is not trivial to calculate the set of logical forms entailed by another.

Advances since Karttunen (1973) have focused on capturing the appropriate empircal details of presupposition projection. However, the basic notion of presupposition as a relation between sentences and logical forms depending on context remains unchanged. Other ideas still in common circulation today are even older, dating back to Frege (1892). One important such idea is the notion that sentences with presuppositions can carry any one of three possible true values: T(rue), F(alse), or (N)either, though

Proceedings of the First International Workshop on Language Cognition and Computational Models, pages 53–62
Santa Fe, New Mexico, United States, August 20, 2018.
https://doi.org/10.18653/v1/P17

this precise naming convention follows Belnap (1979). Such notions remain important through accounts such as the partial account proposed by Beaver and Krahmer (2001).

Beaver and Krahmer's account diverges somewhat from Karttunen's in that it is, in some sense, less pragmatic in that it accounts for presuppositions in the truth conditions of each sentence. For the interface between semantics and pragmatics, Beaver & Krahmer rely only on a valuation function $V : \mathbb{P} \to T, F$, which maps atomic propositions to truth values. Notably, this function's range does not include N, as atomic propositions never carry their own presuppositions. Instead, the determination of presupposition failure falls to the logical form of the sentence and the logical operators on these truth values. As an example, the sentence in (1) could be represented with the logical form in (2), where p represents the proposition "Mary is sad" and q represents the proposition that Bill regrets that Mary is sad (without its presupposition).

(1) Bill regrets that Mary is sad.

(2) $(\partial p \wedge q) \vee \neg \partial p$

For valuations where $V(p) = F$, this formula will evaluate to N, while in valuations where $V(p) = T$ it will evaluate to T or F, depending on the value of $V(q)$. However, once again, this is more complicated in practice than it seems. Actually representing V explicitly in complex situations could require solving some very difficult computational problems. Defining V for all possible propositions is not feasible in a computational environment (including the human brain) unless many values can be predicted from others. However, this amounts to the same problem that I mentioned for Karttunen's model: calculating the set of propositions entailed by another set.

In this paper, I consider this problem more formally, and tackle it by means of a somewhat more restrictive semantics that is incapable of representing complex computational problems, but nonetheless is able to capture the "core" semantics of most concepts. The kinds of reasoning that are necessary for natural language phenomena – in this case, presupposition satisfaction – are within the realm of possibility for this formalism, but more difficult problems never arise. In §2, I further specify the problem of difficult entailment calculations for presupposition. In §3, I re-introduce the formalism of finite state semantics, following work by Rooth (2017). I expand upon this work in §4, introducing finite state semantics for presupposition. Lastly, I discuss the formalism's strengths and weaknesses, consider other possible explanations, and conclude in §5.

A sample implementation of the concepts presented in this paper is available at `https://github.com/thorsonlinguistics/finite-state-presupposition`.

2 Difficult Entailment Problems

Before I consider the general problem of explicit presupposition satisfaction, it may be helpful to consider a few examples where calculating presupposition satisfaction is difficult.

In some contexts, calculating presupposition satisfaction is not possible at all. A simple example occurs with nonce forms and factive verbs, as in (3).

(3) Sam knows that Taylor is a garchank.

Without knowing what a garchank is, an interlocutor cannot determine whether Taylor is one, and thus cannot calculate whether the presupposition is true. However, the interlocutors clearly still have intuitions about the presuppositions of this sentence and it is even possible to construct contexts where the presupposition is clearly satisfied, or where it is clearly *not* satisfied, as in (4) and (5), respectively.

(4) Taylor is a garchank and Sam knows that Taylor is a garchank.

(5) Taylor is not a garchank, but Sam knows that Taylor is a garchank.

Without additional accommodation, (5) is intuitively infelicitous in all contexts, as the factive presupposition that Taylor is a garchank is explicitly contradicted. However, what about (6)? Without knowing what both garchank and quiblet mean, it is again impossible to determine whether the presupposition is satisfied.

(6) Taylor is a quiblet and Sam knows that Taylor is a garchank.

Crucially, presupposition satisfaction is not always *syntactic* (in the logical sense). That is, the fact that *Taylor is not a garchank* ($\neg g$) contradicts *Taylor is a garchank* (g) can be easily determined by the syntactic formulation of the corresponding logical formulas – it is syntactically derivable that any pair of formulas of the form p and $\neg p$ will be contradictory. However, it is *not* syntactically derivable that q and g are contradictory, where q means "Taylor is a quiblet" and g means "Taylor is a garchank." Without further axiomatization to specify that $q \rightarrow \neg g$, the presupposition satisfaction cannot be derived, though speakers still have some intuitions about what it might take for the sentence to be felicitous. When this axiom is introduced, however, it becomes possible to determine that (6) is, in fact, infelicitous.

Consider a more concrete example. Since most native speakers of English know that birds are not mammals, it is fairly intuitive to determine that (8) is an infelicitous utterance in most contexts. However, as described above, this requires knowledge of certain axioms implied by the lexical entries or by the speaker's world knowledge.

(7) Taylor is a cat and Sam knows that Taylor is an mammal.

(8) Taylor is a bird and Sam knows that Taylor is a mammal.

In most cases, this is not actually a problem: the interlocutors are aware of these axioms and can calculate whether they are true in context, whether they are entailed by the linguistic environment, or whether they just aren't known yet (as is the case in (6) without additional information about the meaning of garchank and quiblet).

However, in other cases, it will, in fact *never* be possible to accurately determine whether the presupposition is satisfied. (9), for example, makes reference to the halting problem, which specifies that it is *undecidable* whether an arbitrary program will halt for all possible inputs.

(9) Sam knows that every program on the computer halts.

Can an interlocutor determine whether the presupposition in (9) is satisfied in context? In some contexts, yes. Some programs, of course, do halt, and it may be that all of the programs on the computer do. However, in other contexts, the interlocutors will not be able to determine this fact. Again, the interlocutors still know the conditions under which the sentence is felicitous, but they cannot evaluate this with respect to all possible contexts.

If additional information is added to the scenario, interlocutors may be able to perform additional reasoning. For example, the interlocutors may know that all of the programs on the computer contain 'while' loops that never exit, effectively meaning that none of the programs halt and thus that the presupposition is not satisfied. However, for an arbitrary set of programs, even if that set is fully specified, they cannot determine the felicity of (9).

This poses an important problem. If speakers of natural language perform entailment reasoning in *some* presuppositional contexts, such as (8), but not in others, such as (9), then there is an open question of exactly which sentences fall under which category. Furthermore, since presupposition satisfaction seems to be, in cases like (8), a fairly intuitive, linguistic process, it seems probable that presupposition satisfaction in these cases needs to be calculated fairly quickly. This poses additional problems for cases where presupposition satisfaction *can* be calculated, but requires significant computation.

As an example of a presupposition that is possible, but difficult, to calculate, consider a scenario where the speaker is discussing a checkers game between Sam and Taylor. The speaker may utter the sentence in (10). Actually calculating whether Taylor did make the optimal move in any given situation is possible, but could be quite difficult (Fraenkel et al., 1978). Adding additional discourse information could indicate that Taylor did *not* make the correct move, but the intuition remains that the presupposition might be satisfied – interlocutors do not necessarily know intuitively whether (11) is felicitous.

(10) Taylor knows that she made the optimal move.

(11) Taylor did not queen her piece when she could have, but she knows that she made the optimal move.

In other words, humans only calculate presupposition satisfaction when it is easy. This computation may become easy under various different circumstances, such as when the presupposition is directly stated or once a hard calculation is completed (and accepted by all interlocutors and thus added to the common ground). However, some calculations are *always* easy, such as the contradictory case in (8). Such calculations can be factored into the semantics to account for the intuitive nature of these calculations. I hypothesize that these "easy" calculations are exactly those calculations which can be represented using finite state semantics. Finite state semantics will represent a set of possible worlds for each sentence and will capture the reasoning necessary to capture presupposition satisfaction in some cases, but not in others. In cases where presupposition satisfaction cannot be directly calculated by finite state semantics, the conditions can still be represented and satisfaction can still be characterized.

3 Finite State Semantics

Finite state semantics of the sort that I will utilize here was proposed by Rooth (2017) and itself makes use of the finite state calculus developed by Morhi and Sproat (1996) and Kempe and Karttunen (1996). An implementation of the finite state calculus that could be used for representing finite state semantics is FOMA (Hulden, 2006), which allows for the creation of finite state machines and finite state transducers based on extended regular expressions.

Finite state semantics represents each sentence as a formula of finite state calculus, which can be compiled into a finite state machine (or, in some cases, a finite state transducer). This represents either the set of worlds in which the sentence is true or a relation between worlds (as in the case of questions, following Groenendijk and Stokhof (2002)). I will focus on the case of declarative sentences.

Finite state semantics relies heavily on the notion of centering (Bittner, 2003). As finite state machines are generally capable only of representing sets of strings or binary relations on strings, centering is necessary to distinguish individuals to allow for reference. As an example, the lexical entry for a word such as "cat" would describe the set of worlds in which the center (the most distinguished individual) was a cat. This is done by representing the world as a sequence of individuals, where each individual is defined by a number of properties, including whether the agent has observed it, whether it is the center (or the secondary center, also called the pericenter), and any other characteristics it may have (such as being a cat).

The following definitions show how individuals might be constructed in a model of finite state semantics. There are four kinds of distinguished individuals, represented by the set IDX. These are traces, centers, pericenters, and null, represented by I_0, I_1, I_2, and $I_\varnothing$, respectively. Centers and pericenters are distinguished individuals, with pericenters being secondary (the existence of a pericenter always implies the existence of a center). Null centers are not distinguished and are the default for individuals. Traces are not used in this paper, but are important for the representation of relative clauses. The machine represented by ID is the set of the possible identifiers for elements, which in this case are simple descriptions of the kind of individual being referenced, such as a cat or a dog. In more complex models, these may be much richer representations.

Definition 1 INDIVIDUAL := KNO ID IDX

Definition 2 KNO := $K^+ \cup K^-$

Definition 3 ID := CAT $\cup$ DOG . . .

Definition 4 IDX := $I_0 \cup I_1 \cup I_2 \cup I_\varnothing$

Instances of individuals can be strung together geometrically to create grid-like worlds. For simplicity, I will use only a one-dimensional world, which consists primarily of a string of individuals. The set of all possible worlds is referred to as W. Each proposition is a subset of W indicating the worlds where the proposition is true.

As an example, a simple sentence such as (12) can be translated into finite state semantics using the formula (13). This formula is comparable to the predicate logic formula (14), except that functions such as HASID and INDEF can be reduced to formulas operating directly on finite state machines. The primitive finite state machines in this example include the set of all possible worlds W, as well as worlds

in which the center has the symbol CAT, worlds where the center has the symbol DOG, and worlds where the center is adjacent to the pericenter.

(12) A cat is adjacent to a dog.

(13) $\textsc{Indef}(\textsc{HasId}(\textsc{cat}), \textsc{Indef}(\textsc{HasId}(\textsc{dog}), \textsc{Adj}))$

(14) $\exists x[\textsc{cat}(x) \wedge \exists y[\textsc{dog}(y) \wedge \textsc{adj}(x, y)]]$

Each expression on (13) evaluates to a particular proposition, most of which are intersected together to produce the final proposition, though some additional operations are necessary. For example, the expression $\textsc{HasId}(\textsc{dog})$ indicates the set of worlds in W where the center has the identifier DOG. The expression ADJ represents the set of worlds where the center is adjacent to the pericenter. When ADJ and DOG are intersected, they represent the set of worlds where the center is a dog and is adjacent to the pericenter. The expression $\textsc{Indef}(\textsc{HasId}(\textsc{dog}, \textsc{Adj}))$ further operates on this set to produce the set of worlds where the center is adjacent to a dog (by promoting the pericenter to the center and removing the center). Ultimately, the formula in (13) represents the set of worlds where an individual with the identifier CAT is adjacent to an individual with the identifier DOG.

Of course, it is possible to define much more complex expressions in order to represent other sentences of natural language. In particular, Rooth (2017) defines mechanics for representing intensional semantics and questions using finite state transducers. Rooth also describes how formulas might be produced compositionally from lexical entries using categorial grammars. Crucially, however, finite state semantics provides a compositional means of explicitly representing the set of worlds in which a proposition is true. Reasoning can be introduced by restricting the set using axioms, and some reasoning can even be earned "for free" from the structure of the set of worlds (for example, in this one-dimensional model, it is only possible for an individual to be adjacent to two other individuals).

There is some reasoning, however, that finite state semantics cannot do. For example, attempting to represent sentences such as (15) is difficult. Because the set of worlds where the number of cats and dogs are equal is not a regular set, it cannot be represented using a finite state machine. However, it is still possible to represent, generally speaking, the conditions on the set of worlds. Though Rooth does not discuss this, additional propositions can be easily affixed to the description of each world.

(15) There are the same number of cats as dogs.

Note that no matter the level of computation used, this sort of technique will be necessary for some sentences, such as (9), above. The precise set of worlds where every program halts cannot be fully described by the semantics, so it is necessary to simply state the condition, without fully restricting the set. Note that this will always produce a set of worlds that is *larger* than the "actual" set. As such, this isn't necessarily a problem, it simply indicates a clear boundary between computations that can be carried out in the semantics, and computations that cannot. If finite state semantics is an accurate model of human reasoning, than only finite state computations are performed in the semantics, while other computations are left to higher-level reasoning systems.

However, Rooth's finite state semantics does not provide any mechanism for dealing with presuppositions.

4 Finite State Semantics for Presuppositions

In order to account for presuppositions, I mostly follow Beaver and Krahmer (2001) and use a three-valued logic with Strong Kleene operations. Beaver and Krahmer account for presupposition using a unary presupposition operator ∂ and a binary operator called transplication. The unary presupposition operator has the following truth table.

The transplication operator used by Beaver and Krahmer can be defined using the Strong Kleene connectives $\wedge$, $\vee$, and $\neg$ as well as the partial operator above, such that $\varphi_{\langle\pi\rangle}$ (the proposition φ with the presupposition π) is equivalent to $(\partial\pi \wedge \varphi) \vee \neg\partial\pi$.

As such, there are only a few tasks that need to be undertaken in order to convert Rooth's finite state semantics into finite state semantics with presupposition. First, the basic model needs to be refined in

x	∂ x
T	T
F	N
N	N

Table 1: Unary presupposition

order to account for three-valued logic. Second, the Strong Kleene connectives and the partial operator need to be defined. Finally, these components need to be put together to produce the transplication operator.

The previous model of finite state semantics was incapable of representing three-valued logic because every world in the set was "true", while the set's complement was "false". I account for three-valued logic simply by specifying that every defined world appears in the set and is annotated as either true or false. This produces a set of *valued* worlds W_V instead of a simple set of worlds.[1]

The set of valued worlds can be defined quite trivially from the set W, as shown in Definition 5. Each world in W is simply preceded by a symbol indicating whether it is true or false.

Definition 5 $W_V := (\textsc{true} \cup \textsc{false})\ W$

Defining the Strong Kleene connectives is somewhat less trivial, but can still be done. Strong Kleene "and" is true if both of its arguments are true, and false if either of its arguments are false. Similarly, Strong Kleene "or" is false if both of its arguments are false and true if either one is true. Otherwise, it is neither true nor false. With this in mind, the definitions below can be constructed, where W_t is the set of worlds annotated as "true" and W_f is the set of worlds annotated "false".

Definition 6 $\textsc{KAnd}(X, Y) := W_V \cap ((W_t \cap X \cap Y) \cup (W_f \cap X) \cup (W_f \cap Y))$

Definition 7 $\textsc{KOr}(X, Y) := W_V \cap ((W_f \cap X \cap Y) \cup (W_t \cap X) \cup (W_t \cap Y))$

Strong Kleene negation can be constructed simply by transducing true worlds to false worlds and vice versa. In this definition, $\textsc{Co}(X)$ indicates the co-domain of a binary relation, while Σ indicates the set of all possible symbols in finite state semantics.

Definition 8 $\textsc{KNot}(X) := W_V \cap \textsc{Co}(X \circ ((\textsc{true} \times \textsc{false}) \cup (\textsc{false} \times \textsc{true})\ \Sigma^*))$

Lastly, the partial operator can be defined as the set of valued worlds in W_V where false worlds are removed from the argument – only true worlds are valid.

Definition 9 $\textsc{Presupposition}(X) := W_V \cap (X - W_F)$

Translating the transplication operator at this point is trivial, as all of the operators necessary have already been defined: Strong Kleene connectives and unary presupposition.

Definition 10 $\textsc{Transplicate}(X, Y) := \textsc{KOr}(\textsc{KAnd}(\textsc{Presupposition}(Y), X),$
$\textsc{KNot}(\textsc{Presupposition}(Y)))$

With this tool, it becomes possible to define many presuppositions using finite state semantics, including an extension of Rooth's (2017) intensional semantics for "know" to include a factive presupposition and definite descriptions with uniqueness or maximality presuppositions.

4.1 Factive Presuppositions

Factive presuppositions are introduced by verbs such as *know* in sentences such as (16). The presupposition is satisfied in contexts where the complement of the verb is true.

(16) The agent knows that a cat is adjacent to a dog.

[1] In principle, this actually accounts for a four-valued logic, as there is nothing that prevents a world from being annotated both as a true world and as a false world. Getting rid of this generalization would make the definition of W_V slightly more complicated, and as such I have ignored this possibility. Four-valued logics have also been presented as in some ways "more natural" by, e.g., Herzberger (1973), Karttunen and Peters (1979), and Cooper (1983), which Beaver and Krahmer (2001) note as well.

Assuming that there exists some formula $K(X)$ which indicates that the agent has observed X to be true, it is straightforward to apply the transplication operator to create a factive presupposition, as in (17). For the purposes of this paper, I will only discuss single-agent systems; extending K to a two-place predicate and extending the model to account for multiple agents is left as a future exercise.

(17) TRANSPLICATE(K(X), X)

Rooth (2017) does provide an implementation for $K(X)$, though it requires some modification to work with presuppositions. In particular, the model needs to ensure that any presuppositions that X introduces on its own are projected into the matrix sentence. For example, consider example (18), which contains an embedded presupposition. This sentence is felicitous only where "the cat" can be uniquely identified *and* the cat is adjacent to a dog.

(18) The agent knows that the cat is adjacent to a dog.

Constructing this appropriate definition for $K(X)$ does require a fairly complex definition, but the intuition behind these definitions is simply that the undefined worlds of X are removed. Otherwise, the definition is mostly a straightforward translation of Kripke semantics. R was similarly defined in Rooth (2017); the basic notion behind this relation is that elements which have observed do not vary in the accessible worlds, while other elements are free to vary. This creates an epistemic accessibility relation. K_{base} is the true component of $K(X)$ and is separated from $K(X)$ only in the interest of clarity. UNDEFINEDWORLDS, FALSEWORLDS, and TRUEWORLDS are functions which extract the undefined, false-valued, and true-valued component of a set of valued worlds.

Definition 11 $R := \text{ID} \rightarrow \text{ID} \mid \text{K}^{-}\_$

Definition 12 $K_{base}(X) := \text{TRUE}\ (W - \text{DO}(R \circ \text{FALSEWORLDS}(X)))$

Definition 13 $K(X) := W_V\ \cap\ (K_{base}(X)\ \cup\ (\text{FALSE}\ (W - \text{DEFINEDWORLDS}(K_{base}(X))))) - \text{UNDEFINEDWORLDS}(X)$

These definitions produce the appropriate predictions about presuppositions and presupposition projection. The formula in (19) does not contain any worlds, either in its true or false component, that contain more than one cat or where the cat is not adjacent to the dog.

(19) $K(\text{DEF}(\text{HASID}(\text{CAT}), \text{INDEF}(\text{DOG}, \text{ADJ})))$

4.2 Maximality

As a second example, I consider the case of definite descriptions. The basic notion is, of course, the same: definite descriptions will introduce a formula of the form $\text{TRANSPLICATE}(X, Y)$, where X is the main proposition introduced by the lexical entry and Y is its presupposition. In this case, the presupposition is some form of maximality, indicating that there is a unique collection of individuals that satisfy the restrictor. The other argument of transplication in this case will be a normal application of INDEF. Definites introduce very similar relations when compared to indefinites; they simply have an additional presupposition. The general definition of definites is given below.

Definition 14 $\text{DEF}(X, Y) := \text{TRANSPLICATE}(\text{INDEF}(X, Y),\ \text{UNIQUE}(X))$

There are, of course, a number of theories describing precisely how the presupposition for definites should be constructed (Elbourne, 2013). Many of these theories introduce a simple uniqueness constraint (Kadmon, 1990; Elbourne, 2008; Roberts, 2003). For illustrative purposes in this paper, I will consider only this simple constraint, which only works for singular definites. The implementation of plural definites is given in the supplementary code.

In this case, the intuition behind $\text{UNIQUE}(X)$ is that there can only be one center that satisfies the property X. In worlds where the center currently satisfies X, but a *different* center in the same basic world could also satisfy X, $\text{UNIQUE}(X)$ is not true. A similar intuition can be applied for maximality.

Describing uniqueness requires allowing worlds to (at least temporarily) contain multiple centers and/or multiple pericenters. Of course, this is necessary for describing plurals as well, and so it is not

an unexpected complication. In addition, uniqueness requires the ability to arbitrarily re-assign centers. This is done with the DOREBIND predicate.

Definition 15 $\text{REBIND} := (\text{IDX} \rightarrow \text{IDX}) \cap (W \times W)$

Definition 16 $\text{DOREBIND}(X) := \text{CO}(X \circ \text{REBIND})$

Using DOREBIND, it is again fairly straightforward to define the uniqueness presupposition. The VALUE predicate takes a set of worlds and produces the corresponding set of valued worlds. Again, the undefined worlds of X are removed in order to ensure that presuppositions project properly.

Definition 17 $\text{UNIQUE}(X) := \text{VALUE}(\text{DOREBIND}(\text{TRUEWORLDS}(X) - \text{DOREBIND}(\text{TRUEWORLDS}(X) \cap (\Sigma^* I_1 \Sigma^* I_1 \Sigma^*))),\ X) - \text{UNDEFINEDWORLDS}(X)$

This definition of UNIQUE is used in Definition 14 to construct the lexical entry for the singular definite article. Any reasoning that can be handled by the finite state machine will be automatically calculated in determining the set of valued worlds.

5 Conclusion

By extending Rooth's (2017) finite state semantics to include presupposition, I have also shown how presupposition satisfaction might be calculated in an intelligent system. Crucially, the finite state semantics described here calculates presupposition satisfaction efficiently, without risk of coming across undecidable or computationally expensive problems. There remains some question as to whether finite state semantics is an accurate model of *human* reasoning with respect to presupposition satisfaction and the semantics-pragmatics interface, but it is a *possible* solution.

With this in mind, it is useful to consider the precise predictions that finite state semantics makes for future, empirical work on the psycholinguistics of presupposition satisfaction. Finite state semantics is capable of reasoning about any entailment patterns that are the result of relations between regular sets. Consider the simple, one-dimensional model used in the semantic formulas above. In this model, it is only possible for an element to be adjacent to two other elements. If sentences (20) and (21) are both true (and both refer to the same cat), then the cat cannot *also* be adjacent to a penguin, and the presupposition in (22) should fail according to finite state semantics.

(20) The cat is adjacent to a dog.

(21) The cat is adjacent to a rabbit.

(22) The agent knows that the cat is adjacent to a penguin.

Intuitively, this seems to be true! In a more realistic environment, consider a movie theater, where patrons sit next to each other in a row. A patron can only be sitting next to, at most, two other people, as the people behind and in front of the patron are not usually considered "next to" the guest. Sentence (23) does not seem to be felicitous.

(23) # Sam is sitting next to Taylor and Riley, but Dylan knows that Sam is sitting next to Logan.

On the other hand, there are some contexts that finite state semantics cannot capture. The examples in (9) and (10) are two such cases, for which humans clearly do not calculate the exact set of worlds where the presupposition is satisfied.

Still, there are some cases that are less clear. Finite state semantics is not capable of representing sets that are not regular, including anything higher in the Chomsky hierarchy: context-free languages, context-sensitive languages, or recursively enumerable languages. Constructing natural examples for these sets is difficult, especially as, for more restrictive models, finite state semantics is capable of representing sets that would not be regular in larger models. For example, the set of worlds where (24) is true is not regular. However, if the size of the world is bounded (i.e., no worlds above a particular size are represented in the model), then it can still be represented by finite state semantics.

(24) There are an equal number of cats and dogs.

However, there is additional evidence against a context-free or recursively enumerable semantics, namely that context-free languages are not closed under intersection and recursively enumerable languages are not closed under complement, both of which are used extensively in semantics and reasoning about presuppositions. As such, having a context-free or recursively enumerable semantics as opposed to a regular one would not guarantee cohesion; in some cases, the system would need to rely on more computationally powerful system to represent the desired set at all. Finite state semantics is always capable of producing a set, even if that set is occasionally larger than necessary. Recursively enumerable semantics is especially problematic, as it would require super-Turing computation, thus violating the Church-Turing thesis.

As such, finite state semantics seems to be a reasonable candidate for natural language reasoning for presuppositions, and for many other semantic and pragmatic phenomena besides. Though other solutions to this problem may be possible, especially within the scope of context-sensitive semantics, which would have all of the necessary closure properties, it is generally desirable to make use of the weakest level of computational complexity required, as higher levels of computation are often less efficient. In particular, finite state semantics is capable of representing large sets of possible worlds and performing its calculations in reasonable amounts of time and space, while still representing enough of the semantics to reason about presupposition and provide an interface to higher-level reasoning.

Acknowledgements

Many thanks to the LCCM reviewers, Mats Rooth, Joseph Halpern, and John Foster for their comments.

References

David Beaver and Emiel Krahmer. 2001. A partial account of presupposition projection. *Journal of Logic, Language, and Information*, (10):147–182.

Nuel Belnap, 1979. *A useful four-valued logic*, pages 8–37. Reidel, Dordrecht.

Maria Bittner. 2003. Word order and incremental update. In *Annual Meeting of the Chicago Linguistic Society*, volume 39, pages 634–664. Chicago Linguistic Society.

Robin Cooper. 1983. *Quantification and Syntactic Theory*. Reidel, Dordrecht.

Paul Elbourne. 2008. Demonstratives as individual concepts. *Linguistics and Philosophy*, 31:409–466.

Paul Elbourne. 2013. *Definite Descriptions*. Oxford University Press, Oxford.

A. S. Fraenkel, M. R. Garey, D. S. Johnson, T. Schaefer, and Y. Yesha. 1978. The complexity of checkers on an $N \times N$ board. In *19th International Symposium on Foundations of Computer Science*, pages 55–64, October.

Gottlob Frege. 1892. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, (100):25–50.

Jeroen Groenendijk and Martin Stokhof. 2002. Type-shifting rules and the semantics of interrogatives. In Paul Portner and Barbara H. Partee, editors, *Formal Semantics: The Essential Readings*, pages 421–456. Blackwell.

Hans Herzberger. 1973. Dimensions of truth. *Journal of Philosophical Logic*, (2):535–556.

Mans Hulden. 2006. Finite-state syllabification. In Anssi Yli-Jyrä, Lauri Karttunen, and Juhani Karhumäki, editors, *Finite-State Methods and Natural Language Processing*, volume 4002 of *Lecture Notes in Artificial Intelligence*. Springer.

Nirit Kadmon. 1990. Uniqueness. *Linguistics and Philosophy*, 13:173–324.

Lauri Karttunen and Stanley Peters. 1979. Conventional implicature. In C. Oh and D. Dinneen, editors, *Presupposition*, volume 11 of *Syntax and Semantics*, pages 1–56. Academic Press, New York.

Lauri Karttunen. 1973. Presupposition and linguistic context. *Theoretical Linguistics*, (1):181–194.

André Kempe and Lauri Karttunen. 1996. Parallel replacement in the finite-state calculus. In *Sixteenth International Conference on Computational Linguistics*.

Mehryar Morhi and Richard Sproat. 1996. An efficient compiler for weighted rewrite rules. In *34th Annual Meeting of the Association for Computational Linguistics*.

Craige Roberts. 2003. Uniqueness in definite noun phrases. *Linguistics and Philosophy*, 26:287–350.

Mats Rooth. 2017. Finite state intensional semantics. In *International Conference on Computational Semantics*, Montpellier, September.