Embedding Speech Acts

Manfred Krifka[1]

1. Introduction

The literature on recursion typically focuses on syntactic aspects of recursion. Syntactic rules are recursive if they generate structures in which an expression β occurs within an expression α of the same category c, either directly, as in $\alpha = [_c \ldots [_c \beta\ ] \ldots]$, or indirectly, as in $\alpha = [_c \ldots [_{c'} \ldots [_c \beta\ ] \ldots] \ldots]$. For an example with direct recursion, consider adjectival modification as in [$_N$ *old* [$_N$ *man*]]; for a case with indirect recursion, consider clauses containing complement clauses, as in [$_S$ *John* [$_{VP}$ *thinks* [$_S$ *Mary left*]]]. As the elements of a syntactic category have, by definition, the same internal structure, unlimited recursion is allowed by default, as in [$_N$ *dirty* [$_N$ *old* [$_N$ *man*]]], or [$_S$ *Bill* [$_{VP}$ *suspects* [$_S$ *John* [$_{VP}$ *thinks* [$_S$ *Mary left*]]]]]. If we do not want this, we would have to block recursion explicitly, e.g. by introducing an explicit distinction between the upstairs and the downstairs category, and by rules that make reference to those categories. This would lead to a more complex grammar.

Now, syntax is there for a purpose: to guide the construction of semantic representations. This is done compositionally, that is, the meaning of a complex expression is derived from the meaning of their immediate syntactic parts and the syntactic rule that combines them. Interpretation must be compositional, given that the number of possible expressions is very large or (in case of recursive rules) infinite, as speakers must be able to learn, in a finite and actually quite short time, how to interpret these expressions. If the syntactic rules are recursive, the corresponding compositional semantic rules must be recursive as well. For example, for direct recursion we must allow for functions of a type[2] (σ)σ that take arguments of type σ and deliver values of type σ. For example, we would have to assume, as meaning for an attributive adjective, functions of type **(et)et** like $\lambda P \lambda x[P(x) \wedge \textsc{old}(x)]$, that can be applied to arguments of type **et** like $\lambda x[\textsc{man}(x)]$, and deliver values of type **et** like $\lambda x[\textsc{man}(x) \wedge \textsc{old}(x)]$. This kind of recursivity is standardly assumed in semantics.

Recursive syntax does not mean that every syntactic category is recursive, in the sense that every syntactic category is able to contain expressions of the same category. One kind of systematic exceptions are lexical categories, the categories employed in terminal rules in phrase-structure grammars (in X-bar-syntax, these are categories marked with the superscript 0, e.g. N^0, V^0, etc). This appears to be a natural restriction of recursivity, motivated by the very concept of syntax as a set of rules for the creation of sentences out of words, which themselves are syntactic atoms. Semantically, these are expressions whose meanings have to be learned (which does not preclude that the meanings of some syntactically complex expressions, like idioms, have to be learned as well).

---

1 Thanks to the participants of the Amherst Recursion Conference in 2009, and to Andreas Haida and Sophie Repp. The work reported in this paper has been supported by the DFG (Sonderforschungsbereich 632 "Informations-struktur" and by the Bundesministerium für Bildung und Forschung, Germany (Förderkennzeichen 01UG0711); the responsibilities remain with the author.

2 In naming semantic types, I follow the convention that (σ)τ denotes the type of functions from meanings of type σ to meanings of type τ; if σ is a simple type, parantheses are omitted. Simple types are **e** for entities and **t** for truth values; see below for additional simple types.

But it also has been claimed that certain complex categories that are systematically excluded from recursive syntax. We will consider here so-called root clauses. While clauses can be embedded as complement clauses, adverbial clauses, or relative clauses, it has been claimed by Emonds (1969) and Ross (1973) that there are clauses that (a) allow for certain syntactic configurations, so-called "root transformations," that are not available for other clauses, and (b) do not occur embedded in other clauses, except perhaps in coordinations.

As an example for such root transformations that lead to clauses that cannot be embedded, consider a clause that underwent Left Dislocation:

(1) a. *This room, it really depresses me.*

b. **They put so much furniture in here that* [*this room, it really depresses me*].

Why should there be such a systematic restriction of recursion? Certainly not due to a conceptual necessity of syntax, as in the case of the terminal categories. In phrase-structure grammars, there is no need that the starting symbol, S, cannot occur as the output of a rule. But we may be able to find a reason for this systematic exception of embeddability in semantics, as suggested in Hooper & Thompson (1973): Root clauses are independent assertions – more generally, speech acts – and speech acts generally cannot be embedded.

This is corroborated by those cases in which root clauses actually do embed, and which Hooper & Thompson have pointed out with examples like the following:

(2) *Carl told me that* [*this book, it has receipes in it*].

In such cases, they claim, it is the embedded clause constitutes the main assertion, whereas the embedding clause has the role of a parenthetical expression, in spite of the complementizer *that*. In a sense, there is a mismatch between overt syntax and function; the embedded clause is not embedded after all.

The phenomenon of restrictions of embeddability of clauses and their exceptions grew into an important topic over the years; cf. for example the survey in Heycock (2006). More syntactic phenomena that identified root clauses were found (e.g., modal particles that are specific for root clauses, or verb-second syntax in Germanic languages), and different types of exceptions where identified.

It is quite likely that the various "root transformations" of early generative grammar actually are a mixed bag, and that the various restrictions and their exceptions do not have a uniform analysis. But at least for a substantial subset of cases, a "generalized" version of Hooper & Thompson's explanation seems to be justified. According to this, root clauses have a functional feature that allows them to express assertions or other kind of speech acts, and due to this feature they cannot be embedded, if it were not for those exceptional cases that do allow for the syntactic embedding of speech acts.

In this paper, I will try to argue that this line of argument is semantically sound. In particular, I will try to show what speech acts are, why they typically cannot be embedded, and why they sometimes can. In doing this, I will propose a model-theoretic reconstruction of speech acts, leading to a framework in which both truth-conditional semantics and speech-act theory can be formulated. I will be able to do this with a few examples only, and within a fairly simple, stripped-down semantic theory. Nevertheless, I hope that in this I will go beyond a mere proof-of-concept, and that some of the proposed analyses will be insightful in their own right.

In section (2) I will develop a representation for truth-conditional semantics and for speech acts. The crucial idea, which goes back to Szabolcsi (1982), is that performative acts are not interpreted with respect to word-time indices, but trigger, like other acts, a change in the world in which they are performed. The change that speech acts trigger is a change of the commitments of the participants of conversation. Changes are modelled by a restriction of possibile continuations in a branching-time model. I will show how assertions, questions and commands can be treated within this framework, and I will provide a theory of explicit performatives in which they turn out to be a combination of assertions and other acts. In section (3), I will turn to a number of cases were speech acts appear to be embedded. First, I will discuss logical operations over speech acts: the denegation of speech acts, conjoined speech acts, quantification into speech acts and conditional speech acts. Then I will turn to cases in which speech acts appear in the argument position of predicates, or in which they are modified by other clauses. In the conclusion, #

2. The Formal Representation of Speech Acts

2.1. The Nature of Speech Acts

The literature on speech acts, in particular on the classification of speech acts and the ways how they can be expressed is huge. Here, I will concentrate on the nature of speech acts and on the ways how it should be modelled formally. It is safe to say that there are two broad perspectives on speech acts, as already discussed in Lewis (1970).

One view, most clearly expressed in Stenius (1967) but already present in Frege's distinction between thoughts and judgements (Frege 1879), considers speech acts as communicative actions. They make use of semantic objects like propositions, but transform them to something of a different nature. This view distinguishes between sentence radicals, which denote propositions, and speech acts that are formed when illocutionary operators are applied to sentence radicals. Speech acts are something quite different from semantic objects like entities or propositions; they are moves in a language game in the sense of Wittgenstein (1958). For example, if a speaker asserts *Mary has left* to an addressee, the speaker uses the proposition 'Mary has left' for a particular game. Stenius called this the "report game", which follows the rule "Produce a sentence in its indicative mood only if its sentence-radical is true". An alternative formulation of this rule might be, "the speaker takes on the obligation with respect to the addressee to guarantee that the sentence radical is true, e.g. by providing evidence on request". What is important is that speech acts are actions, not propositions.

Lewis himself favors another view, which he calls "paraphrased performatives"[3]. This view does not make a radical semantic distinction between sentence radicals and speech acts, but considers both to be propositions. In our example, if the speaker asserts *Mary has left* to the addressee, then this can be captured by 'the speaker tells the addressee that Mary has left', which is itself a proposition, hence a semantic object. This leads to the problem that the assertion *Mary has left* would necessarily be true whenever uttered by a speaker to an addressee, and so for all performative sentences. For this reason, Lewis assumes the method of "paraphrased performatives" only for speech acts other than assertions ("declaratives"), and disregards that questions like *Has Mary leaft?,* analyzed as 's asks a whether Mary has left', have a spurious truth value. If we would like to treat speech acts in a homogenous way, we might assume the method of paraphrased perfomatives for assertions as well, and simply disregard the truth value of the whole paraphrase as this is always true. The method of paraphrased performatives is related and probably inspired by the "performative hypothesis" of Katz & Postal (1964) and Ross (1970), which assigns the sentence *Mary has left* a deep

3 The only argument that Lewis gives is that the sentence-radical view would not allow for a treatment of constitutent questions like *Who came?*, and encouragements like *Hurrah for Mary!* This is clearly not the case. Constituent questions can be treated like polarity questions if we assume that their sentence radical denotes a set or propositions or a structured propositions. Encouragements can be seen as speech acts that require a person-denoting referential expression as radical; in our example, *hurrah* can be treated as an illocutionary operator applied to *Mary*.

structure of the form *I tell you* [*Mary has left*]. Such deep structures can be seen as syntactified versions of paraphrased performatives.

The two perspectives on speech acts differ in their consequences concerning the role of semantic representations in syntactic recursion. According to Stenius, speech acts are distinct from regular semantic objects. As regular semantic recursion is defined over entities, truth values, worlds, times, contexts and the functions one can build from them, we should not expect that speech acts can themselves be arguments of semantic operators. Once an illocutionary operator has been applied and has transformed a semantic object into a speech act, there is no chance for it to be embedded again. According to Lewis' method of paraphrased performatives, speech acts are regular semantic objects, and there is no intrinsic reason to assume that speech acts cannot be embedded.

What we find is that speech act embedding occurs, but in restricted ways. I take this as indicating that the Stenius view is right: speech acts are not just propositions, otherwise we would find that they would participate more fully as arguments of semantic operators, like clause-embedding predicates or Boolean operators. But we must explain how the Stenius view can be reconciled with the embedding of speech acts that actually does occur. For this, we first have to develop a theory of speech acts in which they differ from regular semantic objects, but still can be folded back into semantic meanings.

2.2. A dynamic interpretation of speech acts

In this and the next two subsections, a semantic framework will be developed that can account for standard semantic phenomena, and which later will also be able to accomodate performative speech acts.

For the denotational part of the semantics of natural language, the model-theoretic approach of Richard Montague has proved to be extremely fruitful. It is the natural basis on which to build a more general theory of communication that encompasses speech acts. Montague (1973) provided a framework which allows for evaluating the truth value of a sentence with respect to an index (a world and a time). Kaplan (1975) extended this framework by introducting contexts, thus allowing for a principled treatment of deictic expressions referring to the situation of utterance, like speaker and addressee, and world and time of the utterance. The basic explanandum remained the same: the derivation of truth values of sentences. This static picture changed with Stalnaker (1974) and Karttunen (1974), who modelled the communicative impact of expressions as a change of the common ground of speaker and addressee. Kamp (1982), Heim (1982) and Groenendijk & Stokhof (1987) extended this dynamic view, allowing a treatment of pronouns referring to entities mentioned in the prior discourse. The resulting picture is one of a dynamic conversation but a static world: a sentence changes the common ground and the set of available discourse referents, but the world and time of the utterance stays the same. This contrasts with the notion of speech acts as seen by Stenius: Speech acts are not true or false at a world and time, but rather create new facts, after which the world will be different. Communication does not just change the common ground of interlocuturs, it changes the world itself.

It turns out that this view is not quite novel. Szabolcsi (1982), in a paper ahead of its time that was not taken up by semanticists or speech act theorists, sketches exactly this view of speech acts as an index changing device.[4] What follows can be seen as an execution of her idea.

In the following subsection, we will introcude a model frame that provides for an interpretation of regular semantic expressions, but also accomodates speech acts as changes of the world. The model frame will be minimal in the sense that it should illustrate how a dynamic representation of speech

4 Thanks to Hans-Martin Gärtner who directed me towards that paper.

acts would work. We will not attempt to capture discourse referents, or events, or even all aspects of speech acts, like the utterance act (part of the locutionary act) and the perlocutionary act (the achievement of what the speaker intended by the speech act). Rather, we will concentrate here on the essence of speech act, which we call here the "illocutionary act."

What do speech acts change? Speech-act theory has been characterized by two perspectives: (cf. Harnish 2005): First, the idea that in speech acts the speaker expresses some attititude, like a belief or desire; for example, in an assertion, the speaker expresses a desire that the addressee believe the content. Secondly, the idea that in a speech act the speaker takes on certain commitments; for example, in an assertion, the speaker takes on the social commitment that the content of the assertion is true, which involves, for example, the obligation that the speaker is able to provide evidence for the content. I will mostly follow here the second approach (cf. Alston 2000), but I think the theory to be developed here is neutral with respect to the expressed attitudes view or the commitment change view. What is important is that with a speech act, the speaker changes the world – either by making public a certain belief or desire, or a willingness to change certain commitments. For example, after the speech act of an assertion, the world changes to a new state in which the speaker has made clear the desire that the addressee believes the sentence radical of the assertion (which does not necessarily mean that the speaker has achieved that the addressee believes it), or has undertaken the commitment to be liable that the sentence radical is true.

The change in expressed attititudes or commitments is momentaneous, as it doesn't take time. In terms of aspectual classes (cf. Vendler 1967) it is an achievement, like for exampe the verb *arrive*. A linguistic reflex of this is that explicit performatives do not occur in the progressive form, which would express ongoing events. But this refers to the illocutionary act only. The utterance act, which is part of the locutionary act, does take time. We can see the utterance act as an event by which the speaker brings about the state change, using the rules of language. Here, I will disregard utterance acts, and concentrate on illocutionary acts, which are seen as

2.3. Model frames of interpretation

In this section I will propose a model frame at which linguistic expressions, including speech acts, can be interpreted. I will assume a minimal system, for the purpose of a transparent exposition. We will only assume four basic types, and one rule to build up complex types:

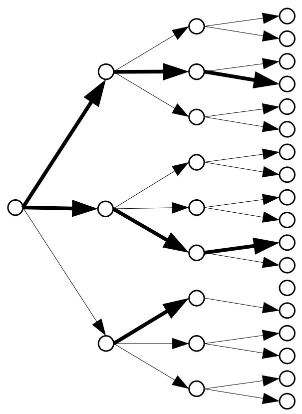
(3) **e**: entities (objects and events)
t: truth values (true and false)
s: indices (world-time-points)
c: contexts (specifying speaker c_s, addressee c_a, time c_t, and current common ground c_g).

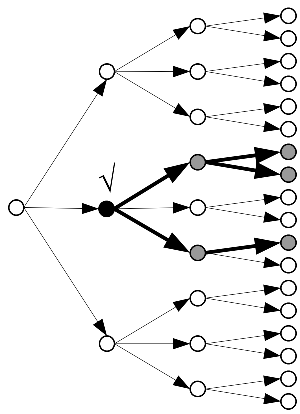
(4) If **σ**, **τ** are types, then **(σ)τ** is a type (functions from **σ**-entities to **τ**-entities);
if **σ** is a simple type, parantheses will be omitted.

A model contains a set of entities **E**. We may introduce sorts of entities, like objects, kind individuals, events, but this is not done here to keep things simple. In particular, events are not expressed explicitly, but we will talk informally about events as changes of states. The truth values are truth and falsity. The indices **I** capture both alternative possible worlds and the flow of time. The atoms (or points) of this set are ordered with respect to an order relation $\leq$ of precedence. This is not a linear order, but for the most part a tree. That is, $\leq$ is transitive, reflexive, and typically left-linear (if $i' \leq i$ and $i'' \leq i$, then either $i' \leq i''$ or $i'' \leq i'$). Left-linearity captures the intuition that the past is fixed but the future is partially indetermined. I will come back to exceptions to left-linearity when we talk about speech act conjunction. We take $<$ to be the corresponding irreflexive order, that is, $i < i'$ iff $i \leq i'$ and $\neg\, i' \leq i$. We call a maximal subset I' of **I** that is linear (i.e., for all $i, i' \in I'$ we have either $i' \leq$

i or i ≤ i′) a “history”. The left-hand side of diagram (5) illustrates a set of world-time indices and three histories in it.

(5) Word-time indices with three histories A rooted interval, or option space.





The right-hand side of (5) illustrates a “rooted interval”. This is a set of indices I′ ∈ **I** that satisfies two conditions. First, I′ has a unique smallest (i.e. a least) element. We call this the “root” of I′, and write √I′. Secondly, I′ is an interval. That is, it holds that if i,i′∈I′, then for all i″∈**I** such that i ≤ i″ ≤ i′ it holds that i″ ∈ I′. We call such rooted intervals “option spaces”. Option spaces capture ways in which the world can develop from the root index. They may cover both the ways how the world can develop following the laws of nature, or the ways how it can develop according to the laws of people, where obligations have to be heeded and desires create certain preferences. In (5), the grey indices represent an option space; the least element, or root, is rendered in black.

The notion of context c used here identifies the speaker c_s and the addressee c_a, as well as a common ground. The common ground collects the ways how the world can be like, according to what the speaker and the addressee have agreed upon. This includes possible pasts, and possible future developments. Contexts also specify the set of accessible discourse referents, and the shared information about the exchange, its purpose, and the mutually agreed upon strategies to develop it – the “common ground management” (cf. Krifka 2008). However, to keep the exposition of the core ideas of this article simple, I will for now just make use of the common ground as the source that provides for a speaker and an addressee.

2.4. Denotational meanings: Propositions

We can distinguish between “denotational” meanings, which give us the reference and truth conditions of expressions with respect to an index (individuals and propositions), and “actional” meanings, which result in the change of option spaces. We first turn to denotational meanings, the traditional realm of semantics.

Denotational meanings are generally functions from contexts to functions from indices, where pure indexicals depend only on the context, and pure non-indexicals depend only on the index of interpretation. If two such meanings $[\![\alpha]\!]$, $[\![\beta]\!]$ are combined, with $[\![\alpha]\!]$ the functor category, the standard meaning combination rule for non-intensional constructions is $\lambda c\lambda i[[\![\alpha]\!](c)(i)([\![\beta]\!](c)(i))]$, that is, the context and the index of the combined meanings is distributed over the constituent meanings. For intensional operators, the argument is the intension, $[\![\beta]\!](c)$. We can give a general type-driven rule for functional applications, as follows:

(6) $[\![\alpha]\!]([\![\beta]\!]) = \lambda c\lambda i[[\![\alpha]\!](c)(i)([\![\beta]\!](c)(i))]$ or $\lambda c\lambda i[[\![\alpha]\!](c)(i)([\![\beta]\!](c))]$,
depending what is well-formed.

Let me illustrate the construction of denotational meanings with a simple example, the sentece *I admired Sue*. I assume that the subject originates in SpecVP and moves to SpecIP, and that the verb originates in V^0 and moves to I^0, where it combines with tense, and expresses agreement with the constituent in SpecIP.

(7) $[\![[_{IP}\ I_1\ [_{I'}\ [_{I0}\ admire_2\text{-PAST}\ [_{VP}\ t_1\ [_{V'}\ [_{V0}\ t_2]\ Sue]]]]]]\!]$
$= \lambda c\lambda i[[\![\text{PAST}]\!](c)(i)([\![admire]\!](c)(i)([\![Sue]\!](c)(i))([\![I]\!](c)(i))]$
with $[\![\text{PAST}]\!] = \lambda c\lambda i\lambda R\exists i[i < c_t \wedge R(c)(i)]$, type **cs(cst)cst**
$[\![admire]\!] = \lambda c\lambda i\lambda y\lambda x[\textsc{admire}(i)(y)(x)]$, type **cseet**
$[\![I]\!] = \lambda c\lambda i[c_s]$, $[\![Sue]\!] = \lambda c\lambda i[\textsc{sue}]$, type **cse**
$= \lambda c\lambda i\exists i[i < c_t \wedge \textsc{admire}(i)(\textsc{sue})(c_s)]$, type **cst**

The result is a function from contexts c to propositions that yield the truth value true for those indices i that are before the time of the context c_t and for which it holds that the speaker of the context c_a admires Sue.

2.5. Speech acts as index changers

We have derived the meaning of a sentence radical, type **cst**. With respect to a given context c, we get a proposition that can be asserted. I assume with Rizzi (1997) that assertion has a syntactic reflex, insofar the IP, the syntactic category for expressions that denote a proposition, is changed to a ForceP, a syntactic category that is interpreted as a speech act, more specifically, an assertion. The category of ForceP is the category of root clauses discussed in section (1).

To express assertional mood, we will make use of an assertion predicate, ASSERT. It takes an index i, an addressee variable y, a proposition p and a speaker variable x, and gives the value true iff at i, x has assertive commitments with respect to the proposition p to the addressee y. This means that x has taken up responsibility to defend the truth of p.

(8) $\textsc{assert}(i)(y)(p)(x) \Leftrightarrow$
at i, the speaker x has assertive commitments
with respect to the proposition p towards the addressee y.

Notice that ASSERT is a state, the state of having assertive commitments. The assertion of a proposition involves a change of states, that is, a change of index from one at which the state of having assertive commitments does not hold to one at which it does. To express this index change, the following notation will be used:

(9) $i \leq i'\ [F[i']]$
'$i \leq i'$, and i′ is maximally like i with the possible exception that F is true of i′.'

We will say that i′ is equal to i "incremented by" the condition F[i′]. If time is dense, the distance between i and i′ is infinitesimal.The performing of an assertion then can be described as follows:

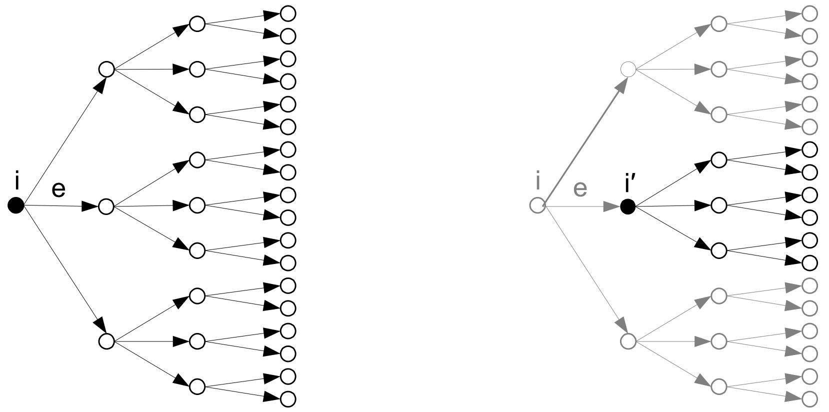
(10) $\lambda i\exists i'[i' \leq i\ [\textsc{assert}(i)(y)(p)(x)]]$

This is true for all indices i that are immediately preceded by an index i′, and i differs from i′ only insofar as at i′, x has assertive commitments with respect to p towards y. This means that the transition from i′ to i models the assertion itself. At i, the assertion is happening and has already happened. Notice that it is not excluded that the assertive commitments already obtains at i – it is not ruled out that the same assertion is made twice. But if this is the case, then it would just be stated that $\textsc{assert}(i)(y)(p)(x)$ holds, and so it can be implicated that $\textsc{assert}(i')(y)(p)(x)$ does not hold for indices i′ preceding i.[5]

5 Alternatively, we could work with a condition $i < i'\ [[F[i']]$, which states that i′ differs from i minimally insofar as i precedes i′, F[i] is false, and F[i′] is true. The weaker proposal in the text is required for the treatment of conditional

The following diagram illustrates this index change. The left-hand side shows the possible courses of history at an index i. In particular, there is one course at which an event e occurs (recall that events are not explicitly represented in the model, but implicitly as changes between states). The right-hand side shows the possible courses of history after that event e has occurred.

(11) Possible courses at an index i Possible courses after event e has occurred.



We can use ASSERT to report about assertions. Arguably, ASSERT is part of the meaning of speech-act verbs like *tell* in examples like *I told you that I admired Sue.* In (12), this example is analyzed up to the VP level, that is, disregarding the tense of the main clause.

(12) $[\![[_{VP}\ \textit{I}\ [_{V''}\ [_{V'}\ \textit{tell}\ [_{V0}\ \textit{you}]]\ [_{CP}\ \textit{that}\ [_{IP}\ \textit{I admired Sue}]]]]]\!]$
$= \lambda c \lambda i[[\![\textit{tell}]\!](c)(i)([\![\textit{you}]\!](c)(i))([\![\textit{that I admired Sue}]\!](c)(i))([\![\textit{I}]\!](c)(i))]$
with $[\![\textit{tell}]\!] = \lambda c \lambda i \lambda y \lambda p \lambda x \exists i'[i' \leq i\ [\textsc{assert}(i)(y)(p(i))(x)]]$, type **cse(sst)et**
$= \lambda c \lambda i \exists i'[i' \leq i\ [\textsc{assert}(i)(c_a)([\![\textit{that I admired Sue}]\!](c)(i))(c_s)]]$

The clausal object of *tell* is an intension of a proposition, not a simple proposition, as we have bound tense in embedded sentences. For example, in *I told you that I admired Sue* the past tense in the embedded sentence is not just interpreted as a simple past (the sentence would be odd if, when uttered in 2011, I want to convey that in 2005 I told you that I would admire Sue in 2008). Rather, past tense is interpreted as relative past PAST_R, referring to the index of telling, where this index is presupposed to be before the time of utterance. For the simultaneous relative past tense we would have the following interpretation (where I do not distinguish between presupposed and proferred information):

(13) $[\![\text{PAST}_R]\!] = \lambda c \lambda i \lambda R \lambda i'[i = i' \wedge i < c_t \wedge R(c)(i)]$
$[\![\textit{that I admire-}\text{PAST}_R\ \textit{Sue}]\!] = \lambda c \lambda i' \lambda i''[i' = i'' \wedge i'' < c_t \wedge \textsc{admire}(i'')(\textsc{sue})(c_s)]$

Plugged in into (12), this yields the following interpretation for the tenseless VP *I tell you that I admired Sue*: A function that takes a context c and an index i, where the index i results from a minimal change from an index i′ such that at i, the speaker c_s has an assertive commitment to the addressee c_a with respect to the proposition that the speaker c_s admires Sue at i, where i must be before the time of context, c_t.

The predicate ASSERT should not be confused with the English speech act predicate *assert*, which expresses an assertion with somewhat higher confidence based on authority. But ASSERT is more general than that; it should capture what all assertions have in common. It is also too general for *tell*, as this verb requires that the assertive commitments came about by a locutionary act that involved speaking. As we do not refer to locutionary acts in our simplified model, this meaning component is disregarded here.

speech acts in section 3.5

(12) showed a descriptive use of ASSERT. But this operator is generally invoked in speech acts of assertion. This is the reason why we have inferences of the following type:

(14) A, to B: *I admired Sue.*
hence: *A told B that he admired Sue.*

Following Szabolcsi (1982), we can represent speech acts like assertions as changes from an input index to an output index. Hence, as speech act, an assertion is not a contextualized proposition (a function from contexts to indices to truth values, type **cst**), but a contextualized index changer (a function from contexts to a functions from indices to indices, type **css**).

I will first present a simple representation of assertions to show how this index change can be expressed; it will be generalized in the next sextion. Take the assertion of the proposition (7) as example. I assume that the ForceP is headed by a syntactic speech act operator ASSERT and has an IP as its complement. The subject is moved to the specifier position of ForceP but is interpreted within the IP, and the tensed verb is moved to the ASSERT head but is interpreted at the head of IP. This leads to the following structure and interpretation.

(15) $[\![[_{\text{ForceP}}\ I_1\ [_{\text{Force}'}\ [_{\text{Force0}}\ \textit{admired}_2\text{-ASSERT}]\ [_{\text{IP}}\ t_1\ t_2\ \textit{Sue}]]]]\!]$
$= \lambda c[[\![\text{ASSERT}]\!](c)([\![[_{\text{IP}} \textit{I admired Sue}]]\!](c))]$
with $[\![\text{ASSERT}]\!] = \lambda c\lambda p\lambda y\lambda x\lambda i\iota i'[i \leq i'\ [\textsc{assert}(i')(p)(y)(x)]]$, type **c(st)eess**
$= \lambda c\lambda y\lambda x\lambda i\iota i'[i \leq i'\ [\textsc{assert}(i')((7)(c))(y)(x)]]$, type **ceess**

The variables y and x will be specified by the addressee and the speaker when this clause is uttered, and i will be specified by the time . Given a context of utterance c, this changes an input index i minimally to that index i′ that differes from i′ only insofar at i′, the speaker of c, x, is committed to defend the truth of the proposition (7)(c) towards the addressee of c, y.[6]

2.6. Speech acts used in conversation

The notion of a speech act that we have derived so far is a "speech act potential" -- it is an device that can be used as a speech act in a concrete situation, that is, in a particular context. Let us look into this more closely and see hwat happens when a speech act potential A of type **ceess** is performed with respect to a particular context.

We make a very simple assumption about contexts: They just specify a speaker, an addressee, and a world-time index of the utterance. Hence, contexts c are triples $\langle c_s, c_a, c_t\rangle$. A speech act will take in these parameters of speaker, addressee, and world-time of the utterance. The result will be a new context in which the speaker and the addressee are the same (we do not model turn-taking here), but the world-time index will be slighly different, because now the speech act has been performed.

(16) $c + A = \langle c_s, c_a, A(c)(c_a)(c_s)(c_t)\rangle$

For example, the execution of (15) at a context c will have the following result:

(17) $c + (15)$
$= \langle c_s, c_a, \iota i'[c_t \leq i'\ [\textsc{assert}(i')((7)(c))(y)(x)]]\rangle$
$= \langle c_s, c_a, \iota i'[c_t \leq i'\ [\textsc{assert}(i')(\lambda i\exists i[i < c_t \wedge \textsc{admire}(i)(\textsc{sue})(c_s)])(c_a)(c_s)]]\rangle$

The world-time index of the input context c_t changes minimally to the index i′ for which it holds that the speaker c_s has assertive commitments towards the addressee with respect to the proposition that the speaker admires Sue at some time before the world-time index of the input context.

6 The speaker and addressee are not specified here as c_s and c_a directly, in order to accomodate for the use in embedded speech acts.

This does not ential, of course, that the asserted proposition actually is true, or taken to be true by the partipants. For this reason, the development of c_t in conversation does not capture the notion of common ground, as the factual information that the participants in conversation ostensibly agree upon. One way to model that is to lift the notion of a context to the notion of a set of contexts, a set of triples $C = \{\langle c_s, c_a, c_t\rangle | ...\}$ as contexts that are constant in their speaker and addressee projection, but differ in their world-time index c_t. Update of a context by a speech act A is pointwise update:

(18) $C + A = \{\langle c_s, c_a, A(c)(c_a)(c_s)(c_t)\rangle | \langle c_s, c_a, c_t\rangle \in C\}$

As before, this captures the information about who said what, and not the content of what has been said. Let us call context updates "credulous" if, in case it was made by an assertion, the content of the assertion is assumed by the participants. Credulous update then can be defined as follows:

(19) $C \mathbin{++} A = \{\langle c_s, c_a, i\rangle | \exists i'[\langle c_a, c_s, i'\rangle \in C \wedge i = A(c)(c_a)(c_s)(i') \wedge$
$\forall p[\neg \textsc{assert}(i')(c_a)(p)(c_s) \wedge \textsc{assert}(i)(c_a)(p)(c_s) \rightarrow p(i)]\}$

This changes every triple $\langle c_a, c_s, i'\rangle$ of the input context C such that its index i′ is changed by the speech act A to i. In addition it holds that if in this change a proposition p was asserted (that is, there was no assertive commitment towards a proposition p at i′ but there is an assertive commitment at i), then the proposition p holds at i. Triples $\langle c_s, c_a, i\rangle$ for which this does not hold are eliminated. Notice that this notion of common ground includes the sources of information, as it does not only hold that p(i) but also that $\textsc{assert}(i)(c_a)(p)(c_s)$. Furthermore, in the history that leads up to an index i the various conversational moves are recorded.

2.7. Speech acts as option space changers

In section 2.5 we saw how speech acts (or rather, speech-act potentials) can be conceived of as semantic operators – as operators that involve an index change. In this section, I will propose a slightly more complex modelling of speech acts, which allows us to treat certain semantic operations over speech acts, like negation. According to this, modelling speech acts do not change indices, but option spaces. For example, if e in (11) is a speech act, it shrinks the input option space (the 31 indices in the left-hand diagram) to an output option space (the 10 remaining indices in the right-side diagram). This is done by the interpretation of ASSERT in the following example:

(20) $[\![[_{\text{ForceP}}\ I_1\ [_{\text{Force}'}\ [_{\text{Force0}}\ \textit{admired}_2\text{-ASSERT}]\ [_{\text{IP}}\ t_1\ t_2\ \textit{Sue}]]]]\!]$
$= \lambda c[[\![\text{ASSERT}]\!](c)([\![[_{\text{IP}}\ \textit{I admired Sue}]]\!](c))]$
with $[\![\text{ASSERT}]\!] = \lambda c\lambda p\lambda y\lambda x\lambda S\lambda i'\in S[\iota i[\surd S \leq i\ [\textsc{assert}(i)(p)(y)(x)]] \leq i']$,
type **c(st)ee(st)st**
$= \lambda c\lambda y\lambda x\lambda S\lambda i'\in S[\iota i[\surd S \leq i\ [\textsc{assert}(i)(y)((7)(c))(x)]] \leq i']$,
type **cee(st)st**

This takes a context c and an input option space S and delivers an output option space – the set of indices i′ that follow the root of S incremented with the assertion by the speaker x to the addressee y of the character (7), evaluated in the context c.

How are speech-act potentials, modelled as functions of type **cee(st)st**, interpreted with respect to a context? It is now convenient to assume that contexts c are tripels consisting of a speaker c_s, an addressee c_a, and an option space c_o, where the world-time index c_t of the context is the root of the option space, $c_t := \surd c_o$.

(21) $c + A = \langle c_s, c_a, A(c)(c_a)(c_s)(c_o)\rangle$

Applied to our example, we get the following result:

(22) $c + (20)$
$= \langle c_s, c_a, \lambda i'{\in}c_o[\iota i[\surd c_o \leq i\ [\textsc{assert}(i)(c_a)(\lambda i\exists i[i < \surd c_o \wedge \textsc{admire}(i)(\textsc{sue})(c_s)])(c_s)]] \leq i']\rangle$

This leads to a reduction of the option space c_o of the input context to those indices i′ that follow the assertion of the sentence radical at the root of c_o, where this sentence radical intself is the proposition that there is an i before the root of c_o at which the speaker admires Sue. Just as before, this guarantees that the proposition (that c_s has assertive commitments towards c_a with respect to the sentence radical) is true at the option state of the output context.

Just as before, this notion of context can be generalized to capture the idea that at any point in conversation there are many candidates of option spaces. To accomodate this, we have to lift the notion of contexts to sets of contexts, as before: We assume sets $C = \{\langle c_s, c_a, c_o\rangle | \ldots\}$ with constant speaker and addressee projection, but different option spaces c_o. Update of a context is then defined as:

(23) $C + A = \{\langle c_s, c_a, A(c)(c_a)(c_s)(c_o)\rangle |\ \langle c_s, c_a, c_o\rangle \in C\}$

Just as in the simple index change theory, we could derive the common ground as containing the content of what has been said, similar as in (19).

2.8. Explicit performatives

We now turn to explicit performative clauses, like *I promise you to come*, which will be compared with descriptive statements, like *I promised you to come*. We take on the descriptive case first. It is based on the following contextualized proposition:

(24) $[\![[_{IP}\ I_1\ [_{I'}\ [_{I''}\ [_{I0}\ \textit{promise}_2\text{-PAST}]\ [_{VP}\ t_1\ [_{V''}\ [_{V'}\ [_{V0}\ t_2]\ [\textit{you}]]\ [_{VP}\ \textit{to come}]]]]]]]\!]$
$= \lambda c\lambda i[[\![\text{PAST}]\!](c)(i)(\lambda c\lambda i[[\![\textit{promise}]\!](c)(i)([\![\textit{you}]\!](c)(i))$
$([\![\textit{to come}]\!](c))([\![I]\!](c)(i))](c)]$
with $[\![\textit{to come}]\!] = \lambda c\lambda i\lambda x[\textsc{come}(i)(x)]$, type **cset**
$[\![\textit{promise}]\!] = \lambda c\lambda i\lambda y\lambda P\lambda x\exists i'[i' \leq i\ [\textsc{promise}(i)(y)(P)(x)]]$, type **cse(cset)et**
$[\![\text{PAST}]\!] = \lambda c\lambda i\lambda R\exists i[i < c_t \wedge R(c)(i)]$, type **cs(cst)cst**
$= \lambda c\lambda i\exists i[i < c_t \wedge \exists i'[i' \leq i\ [\textsc{promise}(i)(c_a)(\lambda i''\lambda x[\textsc{come}(i'')(x)])(c_s)]]]$, type **cst**.

I assume here, somewhat simplifying, that *promise* subcategorizes for a VP, which is interpreted as a property. The formula $\textsc{promise}(i)(y)(P)(x)$ states that at i, x has the commitment towards y to perform P, i.e. to act in such a way as to make P(x) true. The act of promise itself then is a state transition from a state i′ to that state i, where i follows i′ immediately and differs only insofar as in i, the promissive commitment obtains. If this proposition is asserted, we get the following result:

(25) $[\![[_{ForceP}\ I_1\ [_{Force'}\ \textit{promised}_2\text{-ASSERT}\ [_{IP}\ t_1\ t_2\ \textit{you to come}]]]]\!]$
$= \lambda c\lambda y\lambda x\lambda S\lambda i'{\in}S[\iota i[\surd S \leq i\ [\textsc{assert}(i)((24)(c))(y)(x)]] \leq i']$

If this speech-act potential is uttered at a particular context c, we get the following result:

(26) $c + (25)$
$= \langle c_s, c_a, (25)(c)(c_a)(c_s)(c_o)\rangle$
$= \langle c_s, c_a,$
$\lambda i'{\in}c_o[\iota i[\surd c_o \leq i\ [\textsc{assert}(i)(c_a)$
$(\lambda i\exists i[i < \surd c_o \wedge \exists i'[i' \leq i\ [\textsc{promise}(i)(c_a)(\lambda i''\lambda x[\textsc{come}(i'')(x)])(c_s)]]])$
$(c_s))]] \leq i']$

The option space changes is restricted to an option space at whose root there was a change such that the assertive commitments with respect to the sentence radical obtain. The sentence radical itself is interpreted as the proposition that is true iff there is an index i′ before the root of the input option state c_o, the time of the utterance, such that before that time a promise of the speaker to the addressee occurred, a promise to come.

Now we turn to the performative use. Formally, explicit performatives are assertions[7], which should be reflected in their analysis. It is crucial that the verb in explicit performatives is in the present tense, where I assume that it refers to the same index as the assertion itself. That is, it is an instance of a relative tense, just as relative past in (13), from which it differs only insofar as it does not refer to times before the utterance time c_t.

(27) $[\![[_{IP}\ I_1\ [_{I'}\ [_{I''}\ [_{I0}\ \textit{promise}_2\text{-PRES}_R]\ [_{VP}\ t_1\ [_{V''}\ [_{V'}\ [_{V0}\ t_2]\ [\textit{you}]]\ [_{VP}\ \textit{to come}]]]]]]]\!]$
with $[\![\text{PRES}_R]\!] = \lambda c\lambda i\lambda R\lambda i'[i = i' \wedge \neg[i < c_t] \wedge R(c)(i)]$, type **cs(cst)cst**
$= \lambda c\lambda i\lambda i'[i = i' \wedge \neg[i < c_t] \wedge \exists i''[i'' \leq i\ [\textsc{promise}(i)(c_a)([\![\textit{to come}]\!](c))(c_s)]]]$, type **cst**.

Now let's see what happens when this contextualized proposition is asserted:

(28) $[\![[_{ForceP}\ I_1\ [_{Force'}\ \textit{promise}\text{-PRES}_{RS,2}\text{-ASSERT}\ [_{IP}\ t_1\ t_2\ \textit{you to come}]]]]\!]$
$= \lambda c\lambda y\lambda x\lambda S\lambda i'{\in}S[\iota i[\surd S \leq i\ [\textsc{assert}(i)((27)(c)(i))(y)(x)]] \leq i']$
$= \lambda c\lambda y\lambda x\lambda S\lambda i'{\in}S[\iota i[\surd S \leq i\ [\textsc{assert}(i)$
$(\lambda i''[i = i'' \wedge \neg[i < c_t] \wedge$
$\exists i'''[i''' \leq i''\ [\textsc{promise}(i'')(c_a)([\![\textit{to come}]\!](c_s)]]]])(y)(x)]] \leq i']$

Just as with other assertions, the input option space changes minimally from S the set of indices i′ in S such that the root √S is first changed to an i for which it holds that at i, the assertive obligations obtain with respect to the proposition of the sentence radical. Now, this proposition states that at this index i (or rather, at i″, which is equal to i due to the relational present tense) the speaker has the promissive obligations to come, and that the speaker has just at this point obtained them. The only way how the speaker can heed the assertive obligation is that the minimal change that created the assertive obligation also created the permissive obligation. This is exactly how explicit performatives work: The assertion coincides with the promise.[8]

One argument for this analysis involving both an assertive act and a promise act can be derived from the use of the particle *hereby* in such cases. In its basic meaning, *hereby* relates two events as being intimately connected (cf. uses not related to speech acts, as in *The dominant position of English is criticised of destroying smaller languages and hereby destroying cultures*[9]). In the current analysis of explicitly performative utterances, *hereby* relates the "inner" performative event (e.g. the promise) to the "outer" one (the assertion). As an argument for the assumption that the assertive syntax of explicit performatives should be taken serious, consider the fact discussed in Harnish (1988) that performatives can be questioned according to their truth value, just like other assertions: A: *I promise to be there.* B: *Is that true?* Again, this is captured by our analysis, as explicit performatives are assertions.

One might ask what rules out the performative interpretation of assertions like *Sue promises Mary to come*, in which the agent and/or the benefactive role are taken by different referents than the speaker and the addressee of the clause. The reason is that with promises, it is simply implausible to assume a minimal change that is both an assertion by the speaker and a promise by a different person because these two events have different agents. Notice that explicit performatives with passive sentences, in which the agent is imlicit, are fine again, as can be seen with cases like *The dance floor is hereby opened.*

7 Or questions, as in *Do you promise to come?*, or commands, as in *Now, promise to come!* See section 2.9 for these speech acts; however, in this paper I will not treat explicit performatives in these other basic illocutionary types.
8 Due to this property, Koschmieder (1945) called explicit performatives "Koinzidenzfall", the "coincidence case".
9 www.lyseo.edu.ouka.fi/~hequel/linguistic%20imperialism.doc, accessed November 11, 2010.

2.9. Grammaticized speech acts: Assertions, Directives, Questions

In addition to ASSERT, there are two other illocutionary operators that are grammaticalized in many languages. One of these operators is DIRECT, which is present in directive speech acts, and is expressed by the imperative. We can assume that the sentence radical of the imperative is a property referring to the agent of an action. The simplest syntactic analysis seems to be that the verb moves from the subject-less VP to the head of the directive force phrase, and is spelled out as an imperative there.[10]

(29) $\llbracket[_{\text{ForceP}}\ [_{\text{Force}'}\ \textit{come}_1\text{-DIRECT}\ [_{\text{VP}}\ \_\ [_{\text{V}'}\ [_{\text{V0}}\ \text{t}_1\ ]]]]]\rrbracket$
$= \lambda c[\llbracket \text{DIRECT}\rrbracket(c)(\llbracket[_{\text{VP}}\ \_\ [_{\text{V}'}\ [_{\text{V0}}\ \textit{come}\ ]]]\rrbracket(c))]$
with $\llbracket\text{DIRECT}\rrbracket = \lambda c\lambda P\lambda y\lambda x\lambda S\lambda i'\in S[\text{ui}[\surd S \leq i\ [\textsc{direct}(i)(P)]] \leq i']$
$= \lambda c\lambda y\lambda x\lambda S\lambda i'\in S[\text{ui}[\surd S \leq i\ [\textsc{direct}(i)(\lambda i''\lambda x[\textsc{come}(c)(i'')(x)])(y)(x)]] \leq i']$

The formula $\textsc{direct}(i)(P)(y)(x)$ should be understood as: At i, there is an obligation of the addressee y towards the speaker x to perform P, that is, to make it to come about that P(y) becomes true. Notice that the radical of the directive can itself be a an act, as in *Tell me when you will come*, or *Promise me to come.*

The third operator are questions, represented in interrogative clauses. One prominent analysis of questions (Hamblin 1973) assumes that they denote sets of propositions, the set of possible answers. But this is the denotational meaning of questions, their sentence radical. Such sets of propositions can be used to ask a question, but this has to be made explicit by an illocutionary operator, QUEST. On the speech-act level, the speaker of a question puts the addressee under an obligation to react with an assertion of those propositions that are true.

(30) $\llbracket[_{\text{ForceP}}\ \textit{who}_3\ [_{\text{Force}'}\ [_{\text{Force0}}\ \textit{did}_2\text{-QUEST}]\ [_{\text{CP}}\ \text{t}_3\ [_{\text{IP}}\ \textit{you}_1\ [_{\text{I}'}\ [_{\text{I0}}\ \text{t}_2\ ]\ [_{\text{VP}}\ \text{t}_1\ [_{\text{V}'}\ [_{\text{V0}}\ \textit{see}]\ \text{t}_3]]]]]]]\rrbracket$
$= \lambda c[\llbracket\text{QUEST}\rrbracket(c)(\llbracket[_{\text{CP}}\ \textit{who}_3\ [_{\text{IP}}\ \textit{you}_1\ [_{\text{I}'}\ [_{\text{I0}}\ \textit{did}\ ]\ [_{\text{VP}}\ \text{t}_1\ [_{\text{V}'}\ [_{\text{V0}}\ \textit{see}]\ \text{t}_3]]]]]\rrbracket(c))]$
with $\llbracket[_{\text{CP}}\ \textit{who}_3\ [_{\text{IP}}\ \textit{you}_1\ [_{\text{I}'}\ [_{\text{I0}}\ \textit{did}\ ]\ [_{\text{VP}}\ \text{t}_1\ [_{\text{V}'}\ [_{\text{V0}}\ \textit{see}]\ \text{t}_3]]]]]\rrbracket(c)$
$= \lambda p\exists x[\textsc{person}(i) \wedge p = \lambda i'[\textsc{see}(i')(x)(c_a)]]$
$= \lambda c\lambda y\lambda x\lambda S\lambda i'\in S[\text{ui}[\surd S \leq i\ [\textsc{quest}(i)$
$(\lambda p\exists x[\textsc{person}(i) \wedge p = \lambda i'[\textsc{see}(i')(x)(c_a)]])(y)(x)]] \leq i']$

Here, Q is a variable of type **(st)t**, a set of propositions, and $\textsc{quest}(i)(Q)(y)(x)$ is to be understood as: At i, there is an obligation of the addressee y towards the speaker x to assert all and only those propositions p in the set of propositions Q that is assertable, according to the usual rules (i.e., for which the addressee is willing to undergo the obligations that come with an assertion).

There are other illocutionary operators that are grammaticized, like exclamations. However, it should be pointed out that we do not have a large range of grammaticized speech acts. It is true that assertions can be used for a wide variety of speech acts. For example, *I will come – that's a promise!*, or *I promise to come – that's a threat!* I will assume that such speach acts are basically assertions. The derived illocutionary force (a promise, a threat) are just additional specifications of those assertions that we can derive by contextual inferences, or that may be specified with linguistic means, as in *that's a promise* (where the pronoun *that* refers to an assertion event introduced before). This answers one objection of Bierwisch (1980) against introducing speech acts into semantics: We need certain illocutionary operators, like ASSERT, that have grammatical reflexes; we do not need others, like THREAT, or PROMISE, which lack such reflexes, at least in English. This situation is comparable to denotational semantic properties such as 'female' and 'blond': While the

10 Alternatively, there is an immediate IP through which the finite verb moves. Evidence from this comes from number agreement, e.g. German *komm* 'come-IMP.2SG' vs. *kommt* 'come-IMP.2PL' (where this form is identical to the declarative), which is assumed to arise within the IP.

first property has grammatical reflexes in many gender languages (e.g., choice of pronouns, agreement), the second one does not.

3. Embedding of speech acts

3.1. Preliminaries

In the preceding sections, we differentiated between the description of situations and the performing of speech acts. In particular, we distinguished between context-dependent propositions, type **cst**, which are evaluated at an index i, and semantic objects for speech act potentials, type **cee(st)st**, which change the index i (or rather, an option space of which i is the root).

We can now see why speech-act potentials often are resistant against syntactic recursions. They change indices, whereas semantic functions typically are interpreted with respect to a given index. However, we have also derived a way in which we can deal with meanings that operate on index-changing devices. In this section , I will turn to several such semantic functions. In particular, we will deal with denegation and the conjunction of speech acts, and with the quantification into and the conditionalization of speech acts. Then I will discuss predicates that take speech-act potentials as arguments, adjuncts that modify speech acts, and finally appositive relative clauses and parentheticals that express subsidiary speech acts.

A note on presentation: I have proposed a type for speech acts that carries the speaker and the addressee as arguments. In the following, I will sometimes disregard this for reasons of perspicuity, and assume that the speaker and addressee are filled directly by the speaker and addressee of the context. For example, instead of working with a speech act potential $\lambda c\lambda y\lambda x\lambda S\lambda i{\in}S[\surd S \leq i\ [\textsc{assert}(i)[..c..](y)(x)]$, type **cee(st)st**, I will assume a speech act potential $\lambda c\lambda S\lambda i{\in}S[\surd S \leq i\ [\textsc{assert}(i)[..c..](c_a)(c_s)]$, type **c(st)st**.

3.2. Denegation

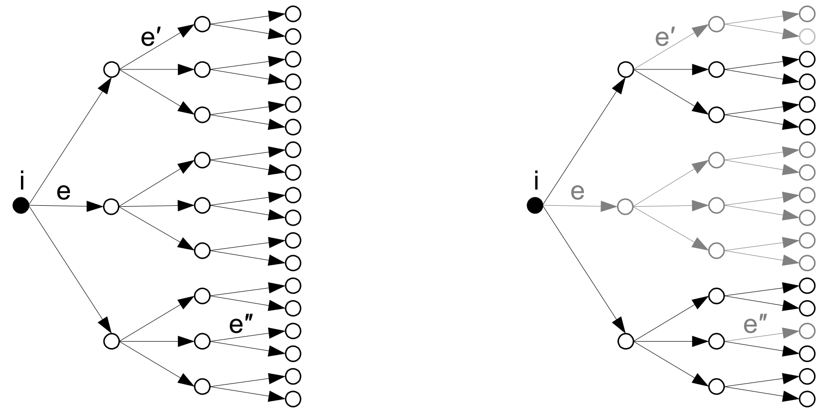
We first consider cases in which speech acts appear to occur in the scope of logical operators. We start with denegation of speech acts, as in *I don't promise to come* (cf. Searle 1969). This can be understood as refraining from performing the speech act *I promise to come* (cf. Hare 1970). It is not a regular speech act, as we cannot use *hereby,* different from usual explicit performatives.

In our current reconstruction of speech acts, denegations can be modelled by excluding all promises to come from the option state S. I assume that negation is c-commanding the Force Phrase:

(31) $[\![[_{\text{NegP}}\ I_1\ [_{\text{Neg}'}\ \textit{don't}\ [_{\text{ForceP}}\ t_1\ [_{\text{Force}'}\ \text{ASSERT}\ [_{\text{IP}}\ t_1\ \textit{promise}\text{-PRES}\ [\textit{to come}]]]]]]]\!]$
$= \lambda c[[\![\text{NEG}]\!](c)([\![[_{\text{ForceP}}\ I\ [_{\text{Force}'}\ \text{ASSERT}\ [_{\text{IP}}\ \textit{promise}\text{-PRES}\ [\textit{to come}]]]]]\!](c))$
with $[\![\text{NEG}]\!](c) = \lambda A\lambda S[S - \bigcup\{S'{\subseteq}S \mid A(S')]$, type **((st)st)(st)st**:
$= \lambda c\lambda S[S - \bigcup\{S'{\subseteq}S \mid [\![[_{\text{ForceP}}\ I\ [_{\text{Force}'}\ [\text{ASSERT}\ [_{\text{IP}}\ t_1\ \textit{promise}\text{-PRES}\ [\textit{to come}]]]]]]\!](c)(S')]]$
$= \lambda c\lambda S[S - \bigcup\{S'{\subseteq}S \mid (25)(c)(S')\}]$

The denegation *I don't promise to come* removes from the input option state S all those indices that belong to option states S′ at which the explicit performative *I promise to come* has been asserted – that is, all those option states S′ at whose root the promise to come has been made. The effect of this can be illustrated in the following diagram, with the input option state on the left, and the output option state on the right. The three events e, e′ and e″ are speech act events in which the speaker of the utterance promises the addressee to come. The indices after those events – these are the indices for which it holds that this promise was made – are removed. Notice that the root of the option state, i, stays the same; what negation does is to restrict possible future developments.

(32) Input option state, with e, e′, e″: promising events. Output option state after denegation of promises.



We can call such acts that do not directly change an index, but have an effect on subsequent indices, "meta speech-acts" (cf. Cohen & Krifka, to appear). But not that, as we have lifted the effect of speech acts from a change of an index to a change of a set of indices, simple speech acts and meta speech-acts are of the same semantic type, **c(st)st**. What distinguishes them, formally, is that regular speech acts change the root of option states, whereas meta speech-acts change option spaces in other ways.

We can execute the same change of the option state by expressing a future assertion, *I will not promise to come*. Future tense states, in a model of branching time, that in all future developments that are deemed possible, there is one time at which the proposition is true. If a future sentence is negated, all possible future developments at which the event predicted by the non-negated future sentence has occurred are eliminated. This results in the same change as the simple denegation of a speech act.

3.3. Conjunction and Disjunction of Speech Acts

Speech acts of the same general type can easily be conjoined, but they generally resist disjunction (cf. Krifka 2001). Take promises as example. While conjunctions like (33)(a) are fine and have a well-defined meaning that is reflected as conjuctions of the sentence radicals, disjunctions like (b) are odd; they suggest that the speaker hasn't made up his mind about what actually should be promised.

(33) a. *I hereby promise you to sing, and I hereby promise you to dance.*
equivalent to: *I hereby promise you to sing and to dance.*

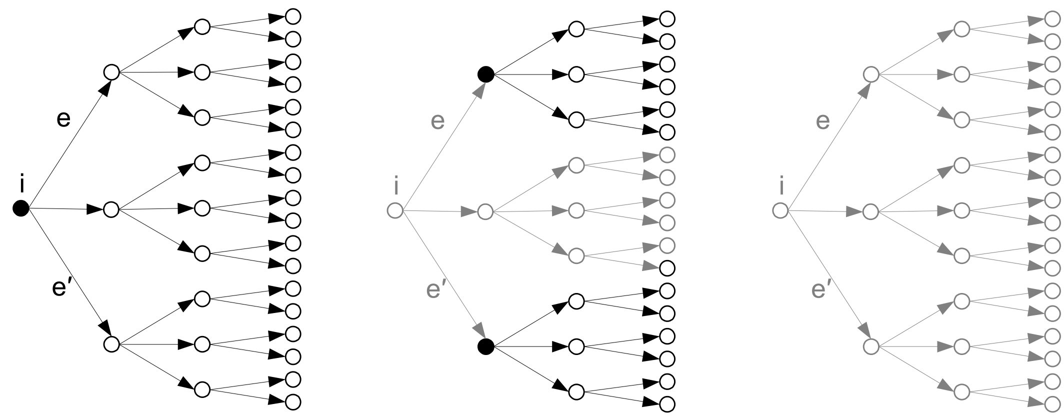
b. *#I hereby promise you to sing, or I hereby promise you to dance.*

This asymmetry of conjunction and disjunction is masked in case an explicit marker of performativity like *hereby* is lacking. In this case, disjunction is interpreted as disjunction of sentence radicals, which is a well-defined operation, as it is just disjunction of propositions.

The asymmetry of conjunction and disjunction is captured by our representation. Conjunction can be interpreted as Boolean or as dynamic. For example, the conjoined proposition *It is raining and it is snowing* can be understood as stating that there is raining and that there is snowing, or as first stating that there is raining, and then stating that there is snowing. Disjunction can only be interpreted as Boolean. The disjoined proposition *It is raining or it is snowing* can only be understood as saying that there is raining or there is snowing.

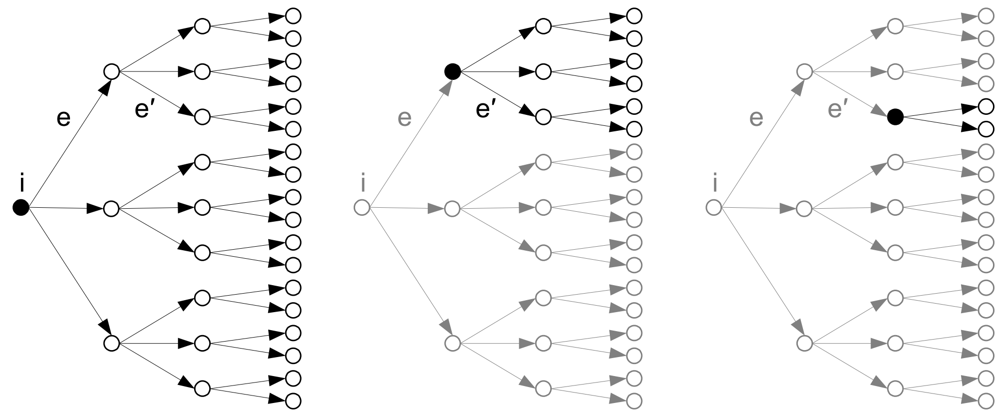
If we conceive of speech acts as changers of option states, as we have done above, then Boolean operations cannot be applied. To see this, consider the following situation, where e and e′ are two speech-act events. The Boolean conjunction of their resulting option state is empty, and the disjunction is not an option state because it does not have a unique root. Lack of a unique root is strongly disfavored, as the obligations of the participants in communications become unclear, in particular after several disjunctive speech acts are performed.

(34) Input option state, with e, e′: speech acts — Disjunction of output option states, as union — Conjunction of output option states, as intersection



In contrast, dynamic conjunction results in a well-behaved output option state:

Input option states, with e, e′: speech acts — Output option state after e — Output option state after e′



This allows for the formulation of speech-act conjunction, as follows:

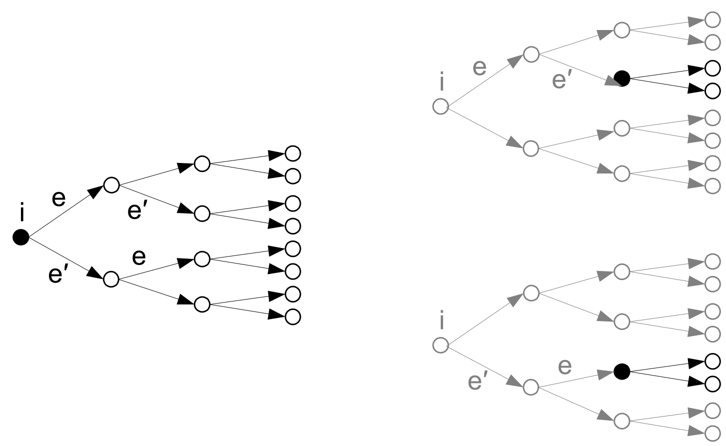
(35) $[\![\textit{and}]\!] = \lambda A \lambda A' \lambda c \lambda S[A'(c)(A(c)(S))]$, type $\mathbf{(c(st)st)(c(st)st)c(st)st}$

As an examle, consider the following:

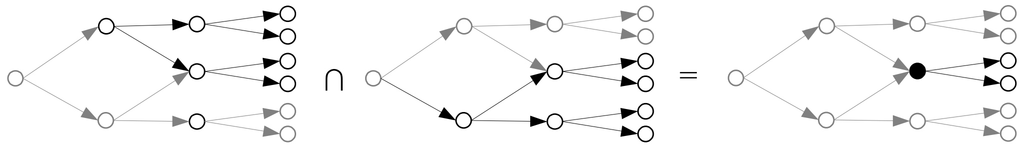
(36) $[\![[_{\text{ForceP}} [_{\text{ForceP}}\ \textit{I}\ \text{ASSERT}\ \textit{promise}\text{-PRES}\ \textit{you to sing}]\ [\textit{and}$
$[_{\text{ForceP}}\ \textit{I}\ \text{ASSERT}\ \textit{promise}\text{-PRES}\ \textit{you to dance}]]]]\!]$
$= [\![\textit{and}]\!]\ ([\![[_{\text{ForceP}}\ \textit{I}\ \text{ASSERT}\ \textit{promise}\text{-PRES}\ \textit{you to sing}]]\!])$
$([\![[_{\text{ForceP}}\ \textit{I}\ \text{ASSERT}\ \textit{promise}\text{-PRES}\ \textit{you to dance}]]\!])$
$= \lambda c \lambda S[[\![[_{\text{ForceP}}\ \textit{I}\ \text{ASSERT}\ \textit{promise}\text{-PRES}\ \textit{you to dance}]]\!](c)$
$([\![[_{\text{ForceP}}\ \textit{I}\ \text{ASSERT}\ \textit{promise}\text{-PRES}\ \textit{you to sing}]]\!](c)(S))]$

One question at this point is whether the order in which the two assertions are made should matter. In our formal representation, it does. If e is a promise to sing, and e′ is a promise to dance, then we have distinct histories when these promises are performed in different orders:

(37) Different results under different order of updates.



(39) Boolean conjunction yields the same result as dynamic conjunction:



This means that we can define speech-act conjunction also als Booleans conjunction:

(40) $[\![and]\!] = \lambda A \lambda A' \lambda c \lambda S[A(c)(S) \cap A'(c)(S)]$, type **c(st)st)(c(st)st)c(st)st**

This is possible, as the resulting set of indices S will be rooted, an option space. In contrast, the corresponding definition of disjunction would not generally result in a rooted option space.

3.4. Quantification into Speech Acts

Speech-act conjunction can be generalized to universal quantification into speech acts, which occurs in cases like the following, under the indicated paraphrase:

(41) *I hereby promise you to do everything you want me to do.*
‘For everything x you want me to do: I hereby promise you to do x.’

We can deal with such cases by assuming that the bound variable x generates a lambda-abstract over speech acts, and that the universal quantifier expresses a generalized conjunction. The order in which the single speech acts are performed does not matter. If we assume that the temporal precedence relation is not left-linear in such cases, then universal quantification over speech acts can be expressed as follows:

(42) $[\![[_{\text{ForceP}}\ [_{\text{DP}}\ \textit{everything you want me to do}]\ \lambda t\ [_{\text{ForceP}}\ \textit{I promise you to do}\ t]]]\!]$
$= \lambda c[[\![[_{\text{DP}}\ \textit{everything you want me to do}]]\!](c)([\![\lambda t\ [_{\text{ForceP}}\ \textit{I promise you to do}\ t]]]\!](c))$
with $[\![\lambda t\ [_{\text{ForceP}}\ \textit{I promise you to do}\ t]]]\!]$
$= \lambda c \lambda P \lambda S\{\mathbf{i'} \in S \mid \surd S < i\ [\textsc{assert}(i)(\lambda i'[i=i' \wedge$
$\textsc{promise}(i')(c_a)(\lambda i''[P(i'')(c_s)])(c_s)])(c_a)(c_s)]] \leq \mathbf{i'}\}$, type **c(set)(st)st**
and $[\![\lambda t\ [_{\text{CP}}\ \textit{you want me to do}\ t]]\!]$
$= \lambda c \lambda P \lambda i[\textsc{want}(i)(\lambda i'[P(i')(c_s)])(c_a)]$, type **c(set)st**
and $[\![\textit{everything}]\!]$
$= \lambda c \lambda \underline{P} \lambda \underline{A} \lambda S \cap \{S' \subseteq S \mid \exists X[\underline{P}(X)(c)(\surd S) \wedge S' = \underline{A}(X)(c)(S)]\}$
$= \lambda c \lambda S[\cap \{S' \subseteq S \mid \exists X[[\![\lambda t[\textit{you want me to do}\ t]]\!](c)(X)(\surd S) \wedge$
$S' = [\![\lambda t\ [_{\text{ForceP}}\ \textit{I promise you to do}\ t]]]\!](c)(X)(S)]$

Here, $\underline{A}$ is a variable over lambda-abstracts of speech acts, and the type of the X is the type of the abstracted variable. Similarly, $\underline{P}$ is a variable over lambda-abstracts of characters. The type of the variable X depends on the constituent abstracted over – here, a one-place property, type **set**. The quantifier *everything* forms the intersection of all option states S′ that result from updating the current option state S with the promise of the speaker to do X, where X varies over what the addressee wants that X does at the current index (the root of S).

Quantification over speech acts is of particular interest in the case of questions. In Krifka (1999, 2001), I have argued that the so-called pair-list reading of questions with quantifiers, as expressed by (43) under the indicated paraphrase, can be dealt with as quantification into speech acts.

(43) *What did every guest bring to the party?*
‘For every guest x: What did x bring to the party?’

As the interpretation of quantification rests on conjunction, we should expect that only universal quantifiers, which are generalized conjunctions, can scope over speech acts. This is justified, as non-universal quantifiers do not lead to a pair-list interpretation. The only interpretation we get in this case is (i), in which the quantifier has scope within the sentence radical of the question; interpretation (ii), where the quantifier scopes over the speech act, is not available.

(44) *What did most guests bring to the party?*
  (i) 'What y is such that for most guests x it holds that x brought y to the party?'
  (ii) # 'For most guests x: What y is such that x brought y to the party?'

In the present and in the preceding subsection we have discussed cases of logical operators like negation and conjunction taking scope over speech acts. In the following section, we will turn to another case of embedding of a speech act by a logical operation, conditionalization.

3.5. Conditional Speech Acts

Austin (1961) has pointed out a certain use of conditionals that cannot be interpreted in the usual, truth-conditional way:

(45) *If you want biscuits, there are some on the side board.*

Notice that the biscuits would be on the side board even if the addressee did not want them. Such examples have been analyzed as involving speech acts in the consequent of the conditional, e.g. by DeRose & Grandy (1999), or by Siegel (2006), who assumed a quantification over "potential" speech acts. It can be taken to mean 'For all indices at which you have the desire for biscuits, the speaker asserts that there are some on the sideboard'. Conditional speech acts are not restricted to assertions, as the following examples with conditionalized questions, commands and explicit performatives show:

(46) a. *If I want biscuits, where can I find them?*
  b. *If she wants biscuits, give her some.*
  c. *If you want biscuits, I promise you that there are some on the side board.*

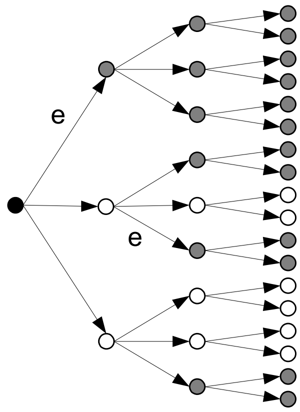
In the present format, Austinian conditionals can be interpreted in a straightforward way. Recall first the definition of $i \leq i'$ [F[i′]] in (9), which requires $i = i'$ if F[i′] is already true at i. We have argued that typically, the index change formula $i \leq i'$ [F[i′]] implicates that it is not the case that F[i], and hence, that $i \neq i'$. But this is a cancellable implicature, which will be relevant for conditional assertion.

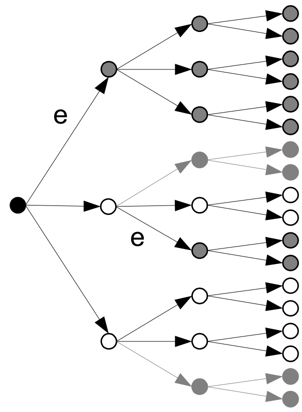
We can implement conditional speech acts following Frank P. Ramsey's insight for "denotational" conditionals, according to which a conditional *if p, q* is true with respect to a "stock of knowledge" if and only if, when p is added to this stock of knowledge, q follows. Speech acts are interpreted not with respect to stocks of knowledge, but with respect to option spaces. Consequently, a conditional speech act *if p, A* should change the input option space such that for all indices at which the condition *p* holds, the speech act *A* is performed[11]. Just as with denegation of speech acts, this will typically not change the root of the input option space, but it will restrain the possible developments of the option space. It is best to make this clear with an illustration.

11 This corresponds to the German term *Vorratsbeschluss,* lit. 'reserve resolution', a term referring to a resolution approved by a committee that becomes operative in case a certain condition holds. It states something like: *If p, then the executive is allowed/is forced to do q*. In English, the term *authorization* is used in this case.

(47) Input option space;
grey: indices where you are hungry,
e: events of asserting that there are bisquits

Output option space
after conditional assertion
if you are hungry, there are biscuits





Only those continuations of the root survive in which it either never holds that you are hungry, or in which it holds that an assertion that there are biscuits was made. This assertion could be made at the root itself, but then the conditional assertion would have the same effect as its unconditional counterpart – *There are biscuits.* Hence, we assume that the model provides for continuations in which the assertions are made some time later, but in any case before the state of the addressee is hungry obtains. Only if the addressee happens to be hungry at the root itself, the root must be changed by the assertion that there are biscuits, and the effect on the option space is the same as with the unconditional assertion.

Formally, this can be captured transparently with the notation borrowed Heim (1992) for conditionals in a dynamic setting. Let us write S + A for the result of applying the speech act A to the option space S (i.e., A(S)), and let us write S + A = same iff S + A = S, that is, updating S with A results in no change of S. The update of S with a conditional speech act *if p, A* then amounts to the following:

(48) $S + \textit{if p, A} = \bigcup \{S' \subseteq S \mid [[S' \cap [\![p]\!]] + [\![A]\!] = \text{same}]\}$

The antecedent proposition p is not a speech act in its own right, but serves to restrict the indices of the option space S. The conditional act if p, A restricts the input option state S in such a way that it contains only the indices of those option states S′ for which it holds that if S′ is restricted to those indices in which the antecedent proposition p is true, then updating this restricted option state with the speech act A does not change it anymore. That is, the communicative effects of A must be already established before. Fehler: Referenz nicht gefunden shows one way how this can be implemented:

(49) $[\![[_{\text{ForceP}}\ [_{\text{CP}}\ \textit{if}\ [_{\text{IP}}\ \textit{you want biscuits}]]$
$\quad [_{\text{Force'}}\ \text{ASSERT}\ [_{\text{IP}}\ \textit{there are some on the side board}]]]]\!]$
$= \lambda c[[\![\textit{if}]\!](c)([\![\textit{you want biscuits}]\!](c)([\![\text{ASSERT}\ \textit{there are some...}]\!](c))]$
with $[\![\textit{if}]\!] = \lambda c\lambda p\lambda A\lambda S \bigcup \{S' \subseteq S \mid A(S' \cap p) = S' \cap p\}$
$= \lambda c\lambda S \bigcup \{S' \subseteq S \mid [\![\text{ASSERT}\ \textit{there are some...}]\!](c)(S' \cap [\![\textit{you want biscuits}]\!](c))$
$\quad = S' \cap [\![\textit{you want biscuits}]\!](c)\}$

The conditional effect resides here in the complementizer *if*, which takes a proposition as first argument, and a speech act as second argument. As for the syntax, the *if*-clause fills the specifier of the ForceP, and the illocutionary operator ASSERT subcategorizes for an IP. Other analyses are possible[12]; here the main point was to show how biscuit conditionals can be interpreted as conditionalized speech acts.

3.6. Speechact-embedding predicates: Assertions

I now turn to cases for which it can be argued that speech acts occur in the argument position of lexical predicates. I have pointed out such cases in Krifka (1999), where I argued that verbs like *wonder* subcategorize for question speech acts. Here, I will first turn to cases in which it appears that lexical predicates take assertions as their argument. Take *say*; this verb can subcategorize for direct speech:

(50) Mary said: "I hate John."

In this use, *say* reports that Mary performed a speech act by a representation of a locutionary act (in particualar, a phonetic act) by which this speech act can be expressed. We tend to require verbatim quotation from direct speech, but this is not always necessary. If Mary used another language, we can translate this into the language used in reporting the speech act. And different genres endorse some liberty in diverting from the wording that has been actually used (cf. the famous case of the speeches in Thukydides' *History of the Pelopponesian War*). The hallmark of direct speech is a complete shift of context; the pronouns referring to speaker and addressee now interpreted with reference to the reported speech situation. The minimal requirement that the sentence *Mary said "..."* is true is that Mary performed a speech act A with respect to a common ground that had the same effects on the commitment states of the speaker Mary and the other interlocutors present as the performing the quoted speech act with the locutionary act "..." would have had. This allows for the possibility that what Mary actually said differs from "..." in certain aspects of its wording.

Putting direct speech aside, there are two further uses of *say* to be considered. In one, *say* subcategorizes for a *that*-clause, in the other, for a root clause:

(51) a. *Mary said that she hates John.*
b. *Mary said she hates John.*

We might argue that (b) is simply a form of (a) that lacks an overt complementizer. But the root-clause property is evident in German, where we have a verb second clause:[13]

(52) a. *Mary sagte, dass sie John hasst.*
b. *Mary sagte, sie hasst John.*

We can model (a) as a case in which *say* subcategorizes a proposition, a sentence radical of an assertion, type **st**, and (b) as a speech act potential, type **ee(st)st**:

(53) a. $\llbracket [_{\mathrm{VP}}\ \mathit{Mary}\ [_{\mathrm{V'}}\ \mathit{say}\ [_{\mathrm{CP}}\ \mathit{that}\ [_{\mathrm{IP}}\ \mathit{she\ hates\ John}]]]] \rrbracket$
$= \lambda c\lambda i[\llbracket \mathit{say} \rrbracket(c)(i)(\llbracket [_{\mathrm{CP}}\ \mathit{that\ she\ hates\ John}] \rrbracket(c))(\llbracket \mathit{Mary} \rrbracket(c)(i))]$
$= \lambda c\lambda i[\textsc{say}(i)(\lambda i'[\textsc{hate}(i')(\textsc{j})(\textsc{m})])(\textsc{m})]$

b. $\llbracket [_{\mathrm{VP}}\ \mathit{Mary}\ [_{\mathrm{V'}}\ \mathit{say}\ [_{\mathrm{ForceP}}\ \mathit{she\ hates\ John}]]] \rrbracket$
$= \lambda c\lambda i[\llbracket \mathit{say} \rrbracket(c)(i)(\llbracket [_{\mathrm{ForceP}}\ \mathit{she\ hates\ John}] \rrbracket(c))(\llbracket \mathit{Mary} \rrbracket(c)(i))]$
$= \lambda c\lambda i[\textsc{say}(i)(\lambda c\lambda y\lambda x\lambda S\lambda i'[\iota i''[\surd S \leq i''\ [\textsc{assert}(i')(\lambda i''[\textsc{hate}(i)(\textsc{j})(\textsc{m})])] \leq i']])]$

12 In German, there is evidence for two distinct constructions. The *if*-clause *wenn du Bisquits willst* (verb-final) can be followed by a verb-initial construction, which can be analyzed as a Force′: $[_{\mathrm{Force'}}\ [_{\mathrm{Force0}}\ \mathit{gibt}_1\text{-ASSERT}]\ [_{\mathrm{IP}}\ \mathit{es\ welche\ in\ der\ Küche}\ t_1]]$. Alternatively, it can be followed by a verb-second construction, which suggests an analysis of the *if*-clause as an adjunct to a ForceP.

13 In addition, there is a subjunctive form (Konjunktiv I) with verb-second but special morphology that can be used: *Maria sagte, dass sie John hasse / sie hasse John.* It generally indicates a speaker different from the speaker of the utterance context.

For the first version of SAY we can give the following meaning, as a first approximation:

(54) $\textsc{say}(i)(p)(x) \Leftrightarrow$ at i, x asserts (by saying) the proposition p.

We can be more precise, as we have introduced an ASSERT operator which describes the commitment state after asserting a proposition:

(55) $\textsc{say}(i)(p)(x) \Leftrightarrow \exists y \forall i'[i < i' \rightarrow \iota i''[i < i'' [\textsc{assert}(i'')(y)(p)(x)]] \leq i']]$

This states that if x "says" proposition p at index i, then the indices i′ that follow i have the following property: First, i is followed by a i″ that differs from i only insofar as x asserts the proposition p to an addressee y. (The addressee could be specified by a *to*-phrase.) Then, all indices that follow i follow that index i″. This means that at i, an event of asserting p by x to some addressee y has occurred.

For the second version of SAY we refer to the reported speech act directly. In a first approximation, we can capture this as in (56), and more precisely as in (57):

(56) $\textsc{say}(i)(A)(x) \Leftrightarrow$ at i, x performs a speech act A,
where A is an assertion.

(57) $\textsc{say}(i)(A)(x) \Leftrightarrow \exists y \forall i'[i < i' \rightarrow \iota i''[i < i'' [A(i'')(y)(x)]] \leq i']]$,
where A is an assertion.

This states that if x "says" speech act A (which must be an assertion) at index i, then i is first followed by an i″ that differs from i only insofar as x performs speech act A to an addressee y (which again could be specified by a *to*-phrase). Furthermore, all other indices i′ that follow i follow this index i″. The selectional restriction for SAY that the speech act A be an assertion excludes cases like **Mary said did John like her*, in contrast to *Mary asked did John like her*, in those varieties that allow for root questions embedded by *ask*.

This shows that a coherent framework can be developed in which a root clause expressing a speech act can be taken as an argument by a lexical predicate. The condition is that the lexical predicate takes an object of the required type. This is the case with *say*, as this predicate describes verbal utterances, including cases in which speech acts have been performed by saying.

Hooper & Thompson (1973) discussed five types of embedding predicates. The verb *say* belongs to their class A, which comprises verbs of saying like *announce, exclaim, vow* etc., and which allow for embedded root clauses. They can be treated similar to *say*, and differ insofar as they express different subtypes of assertions which differ in aspects which can be captured by stating various restrictions on the asserted proposition, on the kind of commitments, and perhaps other aspects (cf. Searle & Vanderveken 1985 and Vanderveken 1990). These verbs can be used with the same meaning in explicit performatives.

We would now have to go through other clause-embedding predicates and check whether the idea that subcategorized root clauses denote speech acts makes sense. Not every case discussed in Hooper & Thompson (1973) will qualify for that. For example, they also list cases like *it's true* or *it's obvious*, which do not report on speech acts. One could perhaps propose that *it's true* subcategorizes for an assertion A expressed by a root clause, with the meaning that the speaker considers A to be assertable. As another example, verbs like *glauben* 'believe' and *denken* 'think' allow for verb-second embedded clauses in German (and also for root-clause phenomena in English). As these verbs do not express speech acts but propositional attititudes, this is in conflict with the idea that root clauses express speech acts. However, we can also assume that propositional attitudes can be characterized by the speech acts that an agent would utter if the agent has a certain propositional attitude. If Mary believes that Bill is at school, then she is willing to assert that Bill is at school, and hence *believe* can subcategorize for such an assertion.

3.7. Question-embedding predicates

I turn here to one class of clause-embedding predicates, those that embedd question clauses. In Krifka (1999), I have argued that predicates that embed questions fall into to classes: Those predicates that embed question sentence radicals, like *know*, and those predicates that embed question speech acts, like *wonder*.[14] This distinction corresponds to the distinction of predicates that embed question extensions vs. intensions in Groenendijk & Stokhof (1984).

(58) a. *John knows who Mary saw.*
b. *John wonders who Mary saw.*

The embedded questions in these examples do not show root-clause syntax with subject-aux inversion (*who did Mary see*). But McCloskey (2005) has pointed out that *wonder*, *ask* and certain other question-embedding verbs, but not verbs like *know* or *find out*, allow for main clause syntax, at least in some varieties of English, especially Irish English. As an example, consider the following quote of James Joyces *Dubliners*:

(59) The baritone was asked what did he think of Mrs. Kearney's conduct.

In German, verbs like *sich fragen* 'wonder' allow for root modals like *denn*:

(60) a. *John weiß, wen Maria* (**denn*) *getroffen hat.*
b. *John fragt sich, wen Maria* (*denn*) *getroffen hat.*

These facts can be taken as evidence that *know* embeds question radicals (in addition to *that*-clauses), and that *wonder* embeds question speech acts. I will illustrate this with the two predicates in question, where I analyze the upstairs clause up to the VP level. First, a case with *know*, applying to a question radical, which is modelled as a set of propositions, following Hamblin (1973).

(61) $[\![[_{VP} \textit{John} [_{V'} \textit{know} [_{CP} \textit{who}_1 [_{IP} \textit{Mary met } t_1]]]]]\!]$
$= \lambda c\lambda i[[\![\textit{know}]\!](c)(i)([\![[_{CP} \textit{who}_1 [_{IP} \textit{Mary met } t_1]]]\!](c))([\![\textit{John}]\!](c)(i))$
with $[\![[_{CP} \textit{who}_1 [_{IP} \textit{Mary met } t_1]]]\!](c)$
$= \lambda p\exists x[\textsc{person}(i) \wedge p = \lambda i'[\textsc{meet}(i')(x)(\textsc{mary})]]$ type **(st)t**

The question-radical embedding *know*, type **cs((st)t)et**, is related to the proposition-embedding *know*, type **cs(st)et**, in the following way, where Q is a variable of type **(st)t**, a set of propositions, as in (30). To "know" a set of propositions at an index i is to know for every true proposition in this set that it is true, and for every false proposition that it is false. This is the rule for complete knowledge; certain weaker forms are possible and have been discussed in the literature.

(62) $[\![\textit{know}]\!](c)(i)(Q)(x)$
$\Leftrightarrow \forall p{\in}P[[p(i) \rightarrow [\![\textit{know}]\!](c)(i)(p)(x)] \wedge [\neg p(i) \rightarrow [\![\textit{know}]\!](c)(i)(\lambda i'\neg p(i'))(x)]]$

Second, consider a case involving *ask*, which embeds question speech acts, and hence is of type **cs(cee(st)st)et**. The following example is supposed to represent the pattern found in Irish English.

(63) $[\![[_{VP} \textit{John} [_{V''} [_{V'} [_{V0} \textit{ask}] [\textit{Sue}]] [_{ForceP} \textit{who}_1 [_{F'} \textit{did}\text{-QUEST} [_{IP} \textit{Mary see } t_1]]]]]]]\!]$
$= \lambda c\lambda i[[\![\textit{ask}]\!](c)(i)([\![\textit{Sue}]\!](c)(i))$
$([\![[_{ForceP} \textit{who}_1 [_{F'} \textit{did}\text{-QUEST} [_{IP} \textit{Mary see } t_1]]]]\!](c))([\![\textit{John}]\!](c)(i))$
with $[\![[_{ForceP} \textit{who}_1 [_{F'} \textit{did}\text{-QUEST} [_{IP} \textit{Mary see } t_1]]]]\!](c)$
$= \lambda y\lambda x\lambda S\lambda i'{\in}S[\iota i[\surd S \leq i\ [\textsc{quest}(i)$
$(\lambda p\exists x[\textsc{person}(i) \wedge p = \lambda i'[\textsc{see}(i')(x)(\textsc{mary})]])(y)(x)]] \leq i']$

14 In Krifka (2001) I assumed that both types of verbs embed question acts, but that verbs like *know* type-shift this question act to the set of true answers. This was designed to handle certain phenomena relating to quantification into embedded questions. Now, and even in 2001, I see advantages of the proposal of Krifka (1999); cf. also McCloskey (2005).

The verb *ask* has the interpretation in (64), and more formally in (65):

(64) ASK(i)(A)(y)(x)
$\Leftrightarrow$ at i, the person x performs a question act of the type A with y as addressee, where A is a question speech act.

(65) ASK(c)(i)(A)(y)(x) $\Leftrightarrow \forall i'[i < i' \rightarrow \iota i''[i < i''\ [\text{ASK}(i'')(y)(A)(x)]] \leq i']$,
where A: a question speech act.

This states that the index i is continued only by such indices i′ for which it holds that they either are identical to an index i″ that differs from i minimally insofar as x and y have the commitments of having performed the question speech act A, or that they follow this index i″.

In the case of *wonder*, no actual speech act need to have occurred. Rather, *wonder* characterizes a state of mind of the subject: The subject is said to be in a cognitive state in which one addresses this question act, typically to oneself (in the German translation, it is *sich fragen*, ‘ask oneself’). This is a speech act that does not need a locutionary (or phonetic) act. We then get the following meaning rule for *wonder*:

(66) $[\![$*wonder*$]\!]$(c)(i)(A)(x)
$\Leftrightarrow$ at i, the person x performs the question act A at i with x as addressee.

Now, *ask* and *wonder* more often embed sentences with non-root syntax, as in *John asked Sue who Mary saw*, or *John wondered who Mary saw*. We can assume a strict mapping between syntactic form and semantic interpretation, and analyze the embedded sentences here as question radicals, that is, sets of propositions Q, just as with *know*. We then would have meaning rules of the following kind, here illustrated for *ask*:

(67) $[\![$*ask*$]\!]$(c)(i)(Q)(y)(x)
$\Leftrightarrow$ at i, the person x performs a speech act with the purpose of wanting to know from y which of the propositions in the set of Q are true at i.

There are other alternatives. We could assume a less strict mapping between syntactic form and interpretation and assume that speech act potentials may be expressed with non-root syntax when embedded. Or we can assume a version of the meaning of $[\![$*ask*$]\!]$ where the effect of the illocutionary operator QUEST is “built into” this meaning, and the resulting meaning subcategorizes for a question radical only.

As McCloskey (2009) has pointed out, even verbs like *know* or *find out* can subcategorize for root clauses in Irish English in certain contexts, like under modals, in questions and imperatives.

(68) a. *Aunt Kate wants to know won’t you carve the goose as usual.* (Joyce, *Dubliners*)
b. *I was dying to find out was he circumcised.* (Joyce, *Ulysses*)
c. *Find out does he take sugar in his tea.*
d. *Do you think will he be re-elected?*

German allows the root-modal *denn* in embedded questions in similar contexts. Hence we cannot just assume as a feature of grammar that verbs like *ask* can embed a speech act potential but verbs like *find out* cannot. Rather, the semantics of the whole construction has to be considered. In all the examples in (68), an interest in the resolution of the question expressed by the embedded question act is expressed, and the question is not presupposed as being resolved already. This seems to be responsible why root-clause syntax is allowed for the embedded questions.

The ramifications of this are still unexplored. In particular, if *know* and *find out* do not allow for speech-act objects, why should *want to know* and *be dying to find out* do? Perhaps these are idioms, but then why does *find out* allow for a speech-act object if it is used in the imperative? McCloskey (2006) assumes that the distribution ultimately depends on semantics: Root-clause embedded questions are possible if the question is unresolved, especially for This explains, for example, why **Does Sally think will he be re-elected?* is worse than (68)(c), as the issue of the question is not an open one for the participants.

3.8. Proxy performatives

There is a use of verbs like *say* in which the main contribution to communication seems to rest on the embedded clause. This was the main motivation that Hooper & Thompson (1973) put forward for embedded root phenomena, but it should be made clear that a similar conversational effect can be achieved with ordinary embedded clauses. Take the following example:

(69) a. *The weather report says that it will be raining.*
     b. *The weather report says it will be raining.*

In typical situations, the main point of these examples is that it will be raining; the weather report is just mentioned as the source of information. We could paraphrase these examples as *It will be raining, according to the weather report*. It is as if the speaker calls up another participant into the communicative situation, the weather report, because he considers this participant to have relevant information. Still, the commitment for the truth of the proposition that it is raining remains with the weather report. The speaker can be made responsible only for the claim that the weather report actually said so. I will call such cases "proxy performatives": The speaker makes a performative statement in lieu of, or proxy for, another person or institution.

We can capture such cases similar to cases of explicit performatives as an index change that requires two changes: First, the assertion by the speaker that the weather report says that it will be raining, and secondly, the assertion by the weather report that it will be raining. This two-fold index change is enforced by the present tense, a feature shared with explicit performatives. This is carried out in the following analysis:

(70) ⟦[$_{\text{ForceP}}$ *The weather report* [$_{\text{ForceP}'}$ ASSERT [$_{\text{IP}}$ t_1 [$_{\text{I}'}$ *say*$_2$-PRES
[$_{\text{VP}}$ t_1 t_2 (*that*) *it will be raining*]]]]]⟧
= λcλyλx[⟦ASSERT⟧(c)(⟦PRES⟧(c)
(⟦[$_{\text{VP}}$ *the weather report say (that) it will be raining*]⟧))]
= λcλyλxλSλi′∈S[ιi[√S ≤ i [ASSERT(i)(y)(x)(λi″[i=i″ ∧
SAY(i″)(⟦(*that*) *it will be raining*⟧(c))(⟦*the weather report*⟧(c)(i″))] ≤ i′]

If x is specified with the speaker c_s and y is specified with the addressee c_a, then this amounts to a function from the option states S into an option state consisting of all indices i′ that follow the root of S changed minimally to an i with the follwing properties: There is at i an assertion of x towards y , and the content of this assertion is true only if the root of S is also changed to one in which the weather report has claimed that it will be raining. Hence, if the assertion by the speaker is true, then the root of S is not only changed to one in which the assertion by the speaker is true, but is also changed to one in which the weather report has asserted that it will be raining.

I regret to inform you that you are hereby dismissed. (Lee).

3.9. Speechact-modifying adverbials

One of the main arguments for the performative analysis were adverbials that appear to attach to the performative prefix, as the following example and its paraphrase suggest (cf. Schreiber 1972, Davison 1973, Sadock 1973 for early literature on the phenomenon, and Mittwoch 1977 for an early critical view).

(71) *Your tie and shirt frankly don't go together*
Frankly, your tie and shirt don't go together
'I tell you frankly [that your tie and shirt don't go together].'

Typically, such adverbs can be used in a descriptive way, as in the following example:

(72) a. *Mary told Bill frankly that his tie and shirt didn't go together.*
b. *Mary frankly told Bill that his tie and shirt didn't go together*.

The two possible positions in the descriptive use lead to meaning differences, the lower position expressing that the way the act was carried out was frank, and the hight position expressing that the fact that the act was carried out at all was frank (cf. McConell-Ginet 1982, Shaer 2003). It seems that it is the second sense that parallels the speechact-related use.

In the current framework, there is no difference in lexical meaning between speech act verbs used descriptively or performatively; there is only a difference in their use: As descriptions, they claim that a speech act of the appropriate type happened with respect to the index of evaluation; as performatives, they change the index of evaluation minimally in such a way that after the change, this speech act has happened. This allows for an analysis of adverbials like *frankly* that accomodates both the descriptive and the performative use.

We start with an example of the descriptive use, in which *frankly* is a preposed adverbial modifier of the VP headed by a speech-act reporting verb (I assume that the clausal argument is extraposed).

(73) $[\![[_{VP} [_{VP} [_{VP}$ *Mary* $[_{V'}$ *frankly* $[_{V'}$ *tell Bill* $t_1]] [_{CP}$ *that* $\Phi]_1]]]\!]$
$= \lambda c \lambda i[[\![\textit{frankly}]\!](c)([\![[_{V'} \textit{ tell Bill that } \Phi]]\!](c))([\![\textit{Mary}]\!](c)(i))]$

What does *frankly* mean? As all lexical meanings, it takes a context c (irrelevant here as *frankly* is not an indexical) and an index i as first arguments. If we treat it as a V′ adverbial, in should then take a property P, type **set**, and map this to another property, which then is applied to the index of evaluation and the subject referent x to yield a truth value. The manner adverb *frankly* subcategorizes for speech acts, and so we can give the following preliminary meaning rule:

(74) $\textsc{frankly}(i)(P)(x)$
$\Leftrightarrow$ x acted in a frank way in performing the speech act P at i.

To act in a frank way in performing a speech act means to act in an open and direct way, possibly breaching certain conversational rules like politeness. that govern the use of speech acts. In our modelling, a speech act consists of a particular index change from an index in which the characteristic commitments of a speech act do not obtain to an index in which they do. Adverbials like *frankly* evaluate this change of indices; i particular, preposed *frankly* states that at the index i, the speaker had a choice of performing the speech act or not performing it, and the choice to perform it was frank.

We have seen in (57) how the reconstruction of speech acts as index changers can be used to capture the descriptive use of speechact-denoting verbs like *say*. Basically, a speech-act proposition like *Mary tell Bill that* Φ is true of an index i iff i has a single immediate successor i′, and i′ differs from i minimally insofar as at i′ Mary has the commitments coming with having asserted to Bill that Φ. The adverb *frankly* evaluates the transition from i to i′, as caused by the speaker, here Mary. We then can give the following interpretation:

(75) $\textsc{frankly}(i)(P)(x)$
$\Leftrightarrow \forall i''[i < i'' \rightarrow \iota i'[i < i' \, [P(i')(x)]] \leq i''] \wedge \textsc{frank}(i, \iota i'[i < i' \, [P(i')(x)]], x)$

This states that $\textsc{frankly}(i)(P)(x)$ holds if i first is succeeded by the index i′ that minimally differs from i insofar x has performed the act P, and for which it also holds that x acted in a frank way to cause the transition from i to the index i′. As i has just one immediate successor, i′, that differs from i just insofar as the speech act is caried out, we can write the second condition more succinctly, as $\textsc{frank}(i)(x)$. When we apply this to our example (73), we get the following result:

(76) $\lambda c\lambda i[\llbracket \textit{frankly} \rrbracket(c)(\llbracket [_{V'}\ \textit{tell Bill that}\ \Phi] \rrbracket(c))(\llbracket \textit{Mary} \rrbracket(c)(i))]$
$\lambda c\lambda i\ [\forall i''[i < i'' \rightarrow \mathrm{u}i'[i < i'\ [\textsc{tell}(i')(\textsc{b})(\llbracket \Phi \rrbracket(c))(\textsc{m})]] \leq i''] \wedge \textsc{frank}(i)(\textsc{m})]$

Let us now turn to a performative use, for which I assume that *frankly* attaches to ForceP′:

(77) $\llbracket [_{\mathrm{ForceP}}\ \textit{I}\ [_{\mathrm{ForceP'}}\ \textit{frankly}\ [_{\mathrm{ForceP'}}\ \mathrm{ASSERT}\ [_{\mathrm{IP}}\ t\ [_{\mathrm{I'}}\ \textit{don't}\ [_{\mathrm{VP}}\ t\ \textit{like Bill}]]]]]] \rrbracket$
$= \lambda c[\llbracket \textit{frankly} \rrbracket(c)(\llbracket \mathrm{ASSERT} \rrbracket(c)(\llbracket [_{\mathrm{IP}}\ \textit{I don't like Bill}] \rrbracket(c)))]$
with $\llbracket \textit{frankly} \rrbracket = \lambda c\lambda A\lambda y\lambda x\lambda S\lambda i{\in}S[A(y)(x)(S)(i) \wedge \textsc{frank}(\surd S)(x)]$,
$\llbracket \mathrm{ASSERT} \rrbracket(c)(\llbracket [_{\mathrm{IP}}\ \textit{I don't like Bill}] \rrbracket(c))$
$= \lambda c\lambda y\lambda x\lambda S\lambda i'{\in}S[\mathrm{u}i[\surd S \leq i\ [\textsc{assert}(i)(y)(\lambda i''\neg[\textsc{like}(i'')(b)(c_s)])(x)]] \leq i']$
$= \lambda c\lambda y\lambda x\lambda S\lambda i'{\in}S[\mathrm{u}i[\surd S \leq i\ [\textsc{assert}(i)(y)(\lambda i''\neg[\textsc{like}(i'')(b)(c_s)])(x)]] \leq i'] \wedge \textsc{frank}(\surd S)(x)]$

When applied to a context c, an addressee y, and a speaker x, this changes an input option state S to an output option state $\lambda i'{\in}S[...]$ for which it holds that at the root of S, an assertion by the speaker to the addressee is made with the content that the speaker does not like Bill. In addition, this transition from the root of S to the following state is a frank action by the speaker. Notice that this part is not claimed to be true, as the content of the assertion is. Rather, the speaker allows only for such continuations of the root of S where this action is frank. This captures the fact that speech-act adverbials cannot be denied or questioned.

3.10. Speechact-modifying adverbial clauses

In addition to adverbials like *frankly* or adverbial phrases like *to be frank*, there are adverbial clauses that are interpreted as modifying speech acts (cf. Sadock 1974). Consider the *because* clause in (78) and contrast them with a case in which *because* expresses a causal relation between propositions, (79).

(78) *Peter wants to be noticed, because he is whistling.*

(79) *Peter is whistling because he wants to be noticed.*

(79) states that the reason why Peter is whistling is that he wants to be noticed; in (78) the speaker justifies the assertion that Peter wants to be noticed by stating that he is whistling.

The interpretation of *because*-clauses in performing a speech act, as in (78), can be dealt with in similar ways as with *frankly* in examples like (77). However, there are important differences. First, the *because* clause appears to be a speech act in its own right. One reason is that it allows for speech-act adverbials:

(80) *Peter is unhappy because quite frankly, few people like him.*

Also, notice that in German, speechact-related clauses exhibit verb second syntax:

(81) *Peter muss glücklich sein, denn / weil er summt ein Liedchen.*
‘Peter must be happy, because he is humming a song.’

Second, while speech-act adverbials and propositional *because* clauses can occur sentence-initially, speechact-modifying adverbial clauses cannot:

(82) a. *Frankly, Peter is unhappy.*
b. *Because he wants to be noticed, Peter is whistling.*
c. **Because he is whistling, Peter wants to be noticed.* (in intended interpretation)

This suggests a slightly different treatment from adverbials like *frankly*. Sentences like (78), (80) and (81) appear to denote a conjunction of two speech acts. First, the speech act of the main clause is performed; this is followed by another speech act that justifies the first. This captures both the root-clause properties of the *because* clause and the order restriction.

As the speech act expressed by the *because* clause carries a certain function with respect to the other speech act, it is not the main claim of the utterance that the speaker highlights. Relations of this type have been modelled in Rhetorical Structure Theory (RST) in asymmetric Nucleus-Satellite relations (Mann & Thompson 1987, Taboada & Mann 2006) and also in Segmented Discourse Representation Theory (SDRT, Lascarides & Asher 2003). I cannot attempt to introduce the concepts developed there in our current theoretical framework, but I would like to claim that the rhetorical relations of RST and SDRT can be seen as relations between speech acts and groups of speech acts. The locus of the so-called “presentational” rhetorical relations is not so much that factual claims are being made, but rather that motivations are given why certain speech acts are being made. Consequently, they belong rather to the management of the common ground than to the content of the common ground.

With our current means, we can model speechact-related *because* clauses as follows: First the main speech act A is performed (e.g., the assertion that Peter wants to be noticed). This creates the index change that comes with the performance of speech acts. Then another speech act B is made about this change, an assertion with the content that the performance of speech act A is justified because some other fact holds – in our example, that it is justified to assert that Peter wants to be noticed because he is whistling. This second speech act is an assertion, but its content cannot be questioned or negated by the addressee, as it is the speaker who has the sole authority to give the justification why he or she performed the first speech act. The speaker would fail, of course, if the reason of justification given is not true – e.g., if Peter were not whistling. Hence this proposition is somehow downtoned compared to the proposition of the main speech act, A.

These considerations lead to the following proposal:

(83) $[\![[_{\text{ForceP}} [_{\text{ForceP}}\ \textit{Peter}_1\ [_{\text{Force}'}\ \text{ASSERT}\ [_{\text{IP}}\ t_1\ \textit{wants to be noticed}]]]$
$[_{\text{ForceP}}\ \textit{because}\ [_{\text{Force}'}\ \text{ASSERT}\ [_{\text{IP}}\ \textit{he is whistling}]]]]]\!]$
$= \lambda c \lambda S \lambda y \lambda x \lambda i{\in}S \exists S'[$
$S' = [\![[_{\text{ForceP}}\ \textit{Peter}_1\ [_{\text{Force}'}\ \text{ASSERT}\ [_{\text{IP}}\ t_1\ \textit{wants to be noticed}]]]]\!](c)(S)(y)(x)$
$\wedge\ [\![[_{\text{ForceP}}\ \textit{because}\ [_{\text{Force}'}\ \text{ASSERT}\ [_{\text{IP}}\ \textit{he is whistling}]]]]\!](c)\mathbf{(S,S')}(S')(y)(x)(i)]$

The two speech acts are conjoined dynamically: First, the input option state S is changed by the first speech act to an output option state S′, which is then fed into the second speech act. For the second speech act I assume that *because* is in the Spec of ForceP, which has the interpretational effect that the second speech act gives a justification for the speech act that has been performed immediately before. For this, we need to refer to the initial option space S and to the option space after the first speech act S′, as this is the semantic effect of performing the first speech act. This is the boldfaced argument of the *because* clause. Notice that we cannot just refer to the roots of these option spaces, as we find justifications also in cases of speech acts that do not change the root, as in denegations:

(84) *I don't promise to come because I don't know whether I will be able to.*

Let us be more specific about the second speech act. The presence of *because* in SpecForceP requires that this speech act is about another speech act that has been uttered immediately before. Hence the *because* clause has an argument referring to a speech act, here represented as a pair of two option spaces. Also, speaker and addressee are necessarily identical to speaker and hearer of that speech act. In the following, the trace T_2 refers to a pair of option spaces.

(85) $[\![\lambda T_2\ [_{ForceP}\ \textit{because}_1\ [_{Force'}\ \text{ASSERT}\ [_{CP}\ T_2\ [_{C'}\ t_1\ [_{IP}\ \textit{he is whistling}]]]]]]\!]$
$= \lambda c\lambda(S,S')\lambda S'\lambda y\lambda x\lambda i[[\![\text{ASSERT}]\!](c)$
$([\![\lambda T_2[_{CP}\ T_2\ [_{C'}\ \textit{because}\ [_{IP}\ \textit{he is whistling}]]]\!](c)(S,S')(S)(y)(x)(i)]$

Finally, the CP expresses the following proposition:

(86) $[\![\lambda T_2[_{CP}\ T_2\ [_{C'}\ \textit{because}\ [_{IP}\ \textit{he is whistling}]]]\!](c)(S,S')$
$= \lambda i[[\![\textit{because}]\!](i)(S,S')([\![\textit{he is whistling}]\!](c))]$

This states that a reason why the speech act represented in the option space pair S and S′ as input and output is due to the proposition 'he is whistling', which entails, among others, that this proposition is true at i.

The overall change of the input option space S accomplished by (83) is then that first it is asserted that Peter wants to be noticed, leading to an output option space S′. Then, it is asserted by the same speaker to the same addressee that the reason for this assertion (i.e. this change of S to S′) is that Peter is whistling. The overall output option space will also contain this second assertion, which entails that Peter is indeed whistling. But as it is only the speaker who can judge the truth or falsity of whether Peter's whistling indeed was a justification for asserting that Peter wants to be noticed, this second assertion, and its entailed proposition, will be less prominent in the overall conversational move.

3.11. Appositive relative clauses and other parentheticals

Appositive, or non-restrictive, relative clauses are clearly embedded root clauses, and hence candidates for embedded speech acts, according to the theory developed here. They allow for speech-act related adverbials, like *frankly.*

(87) *John, who I frankly don't like, wants to visit me tomorrow.*

Such cases can be interpreted as involving two conjoined speech acts. In (87), it is first asserted that John will visit me tomorrow, and then it is asserted that I don't like John, which is marked as a frank assertion. The assertion made in the relative clause has less communicative weight, but it is not presupposed, as it can carry new information.

What is important is that appositive relative clauses stand in a certain rhetorical relation to the main clause, otherwise infelicity results. Holler (2005) distinguishes between two kinds of non-restrictive relative clauses that differ in the type of the rhetorical relation to the main clause. The relative clause in (87) is subordinating, elaborating on the event reported in the main clause, whereas the relative clause in (88) is coordinating, reporting a subsequent event.

(88) *Bill was looking for a phone, which he finally found.*

For the purpose of this article, it is of importance that we have developed a system of representation that allows us to represent the conjunction of speech acts, as well as

In addition, for a

We can interpret such cases as two separate speech acts. In the current framework we can treat such cases by assuming that the appositive relative clause creates a condition for the context. In the case of (87), this condition is that there is a (frank) assertion in the context that the speaker of the context does not like John. This condition is accomodated. To implement this, I make use of the lambda notation λx. <u>Condition for x</u> [Value of x], where the condition for x specifies a condition for the domain of the function. In semantic composition, such conditions are generally projected from subordinated constituents to superordinated constituents. We then can analyze a simplified version of (87) as follows:

(89) $[\![[\textit{who}_1\ [_{\text{ForceP}}\ \text{A}\ [_{\text{CP}}\ \textit{I don't like}\ t_1]]]]\!]$
$= \lambda x_1 \lambda c.\underline{\exists a[a = \text{Assert}(c)(\lambda i[i \sim c \wedge \neg\ \text{like}(i)(x_1)(S(c))])]}\ \lambda i[x_1(c)(i)]$

$[\![\textit{John}]\!] = \lambda c \lambda i[\text{john}]$

$[\![[_{\text{DP}}\ \textit{John}\ [\textit{who}_1\ [_{\text{ForceP}}\ \text{A}\ [_{\text{CP}}\ \textit{I don't like}\ t_1]]]]\!]$
$= \lambda x_1 \lambda c.\exists a[a = \text{Assert}(c)(\lambda i[i \sim c \wedge \neg\ \text{like}(i)(x_1(c)(i))(S(c))])]\ \lambda i[x_1(c)(i)](\lambda c \lambda i[\text{john}])$
$= \lambda c.\underline{\exists a[a = \text{Assert}(c)(\lambda i[i \sim c \wedge \neg\ \text{like}(i)(\text{john})(S(c))])]}\ \lambda i[\text{john}]$

I assume that the relative pronoun of appositive relative clauses scopes over the illocutionary operator, which creates a case of binding into a speech act. The relative construction forces the assertion to be interpreted as a condition on the context c, namely that there is an act a that is identical to the assertion in c that the speaker of c does not like John. (Here, $i \sim c$ says that the world and time components of i and c are the same). The effect of this is that *John, who I don't like* refers to John, with the requirement that the context contains an assetion that the speaker does not like John. This is a context presupposition that will be accomodated by the addressee.

In the next step, the complex DP with the appositive relative clause is integrated into the main clause:

(90) $[\![\textit{will visit me}]\!] = \lambda c \lambda i \lambda x[c < i \wedge \text{visit}(i)(S(c))(x(c)(i))]$

$[\![[_{\text{CP}}[_{\text{DP}}\ \textit{John}\ [\textit{who}_1\ [_{\text{ForceP}}\ \text{A}\ [_{\text{CP}}\ \textit{I don't like}\ t_1]]]]\ [\textit{will visit me}]]]\!]$
$= \lambda c \lambda i[[\![\textit{will visit me}]\!](c)(i)([\![[_{\text{DP}}\ \textit{John}\ [\textit{who}_1\ [_{\text{ForceP}}\ \text{A}\ [_{\text{CP}}\ \textit{I don't like}\ t_1]]]]\!](c)(i))$
$= \lambda c.\underline{\exists a[a = \text{Assert}(c)(\lambda i[i \sim c \wedge \neg\ \text{like}(i)(\text{john})(S(c))])]}\ \lambda i[c < i \wedge \text{visit}(i)(S(c))(\text{john})]$

The contextual presupposition of the subject DP becomes a presupposition of the clause. The resulting character is itself used as an assertion:

(91) $[\![\text{A}\ [_{\text{CP}}\ [_{\text{DP}}\textit{John}\ [\textit{who I don't like}]]\ [\textit{will visit me}]]]\!]$
$= \lambda c.\underline{\exists a[a = \text{Assert}(c)(\lambda i[i \sim c \wedge \neg\ \text{like}(i)(\text{john})(S(c))])]}$
$\quad \text{Assert}(c)(\lambda i[c < i \wedge \text{visit}(i)(S(c))(\text{john})])$

This is a speech act of the following type: It can be uttered in contexts in which there is an assertion that the speaker of c does not like John; if this condition is satisfied, the speech act itself is an assertion that John will visit the speaker of the context.

This example illustrates one way in which speech acts can be integrated into the recursive semantic component. Meanings are dependent on contexts, and contexts can be restricted to contain particlar speech acts.

As pointed out by Holler (2005), there is generally a rhetorical relation between the speech act of the main clause and the speech act of the appositve relative clause. In the case at hand, the relative clause delivers some background information. This relation is not specified explicitly in our representation, but may be captured by a general requirement: If we perfom a speech act with a contextual presupposition that there is another speech act, there must be some rhetorical relation between the two speech acts.

4.