Utrecht University
March 12, 2010

Embedding Speech Acts

Manfred Krifka
Institut für deutsche Sprache und Linguistik, Humboldt-Universität zu Berlin
Zentrum für Allgemeine Sprachwissenschaft (ZAS) Berlin
krifka@rz.hu-berlin.de

1. Introduction

Two views of speech acts:

A. Speech acts are of a different nature than propositions and other regular semantic objects. Hence they cannot be constituents of recursive semantics.
Consequence: There are no embedded speech acts (Stenius 1967).

B. Speech acts are propositions (Lewis 1970, Vanderveken 1990). Hence it should be possible that they are constituents in recursive semantics.
Consequence: There could be embedded speech acts.

Are there embedded speech acts?

It has often been claimed: No. But if we take evidence like speech-act related adverbials (e.g. *frankly, hereby*) and modal particles (e.g. German *wohl*, *denn*) serious, there are:

(1) a. *He said that he frankly doesn't want on-line dealers anyway,*
b. *Trevor would've been a MUCH better choice than Ryan Egghold, who I frankly don't see any appeal in whatsoever.*
c. *I regret that I have to inform you that you are hereby laid off. (Lee 1975).*

(2) a. *Maria denkt, dass Peter wohl noch kommen wird.*
b. *Sie will wissen, wann er denn kommen wird.*

Yet the occurrence of such clauses is very restricted:

(3) a. *If he *frankly doesn't want on-line dealers anyway, he shouldn't buy things on the internet.*
b. *Trevor would have been a better choice than a candidate who I *frankly think is unqualified.*

(4) a. *Falls Peter *wohl kommt, dann gib ihm ein Handtuch.*
b. *Sie weiß, wann er *denn kommen wird.*

Cf. Haegeman (2007) for root clause phenomena in certain "peripheral" adverbial clauses; Coniglio (2009) for modal particles in dependent clauses.

Point to be made: Even under the assumption that the A view is right, speech acts can be part of semantic recursion and are licensed in certain contexts.
This will be illustrated with a number of examples, and a preliminary representation format for embedding speech acts will be given.

2. The Frege/Stenius view of speech acts

Illocutionary vs. truth-conditional level

Frege (1879): Distinction between a "thought" (= propositon) and "judgement"; formal notation: —Φ proposition, |—Φ judgement; | as speech act operator for assertion (there is also an operator || for definitions).

Stenius (1967): Distinction between "sentence radical" (proposition) and "mood" (illocutionary operator) that applies to the sentence radical; the result is a move in a language game.

For example, assertion is related to the "report game" governed by the rule: "Produce a sentence in its indicative mood only if its sentence-radical is true."

(5) Sentence radical: *John arrived* – Denotes a proposition, semantic type: **st**.
Assertion: *John arrived.* – Is a move in a language game, semantic type?

What are speech acts?

A general view: With speech acts, speakers change social obligations, attitudes, or relations.

- With asserting Φ, speaker wants to make Φ part of the common ground; entails that speaker guarantees that Φ is true, and would bear social consequences if it turns out to be false, especially if speaker knew that Φ was false.
- With promising Φ, speaker undertakes the obligation to behave in a way to make Φ true.
- With commanding Φ, speaker puts addressee under the obligation to make Φ true.
- With expressing Φ (e.g., *How beautiful!*), speaker makes public a particular attitude, from which certain social expectations follow.
- With declaring Φ (e.g. *You are fired.*), speaker changes social relations.

Changes of states are **events**. Hence, speech acts are a particular sort of social events. This captures the **illocutionary** aspect of speech acts.

The change of social states in speech acts is by definition at least partly caused by a linguistic act. This is the **locutionary** event.

Hence, speech acts are complex events, consisting at least of a locutionary and an illocutionary part (cf. Austin 1962).

Embedding of Speech Acts

In event semantics, events are part of the semantic ontology, and hence part of the truth-conditional part of semantics. If speech acts are a particular kind of events, then they should be able to participate in semantic recursion.

3. Formal representation of speech acts

Type system

(6) **e**: entities
o: objects (a subtype of entities)
v: events (a subtype of entities)
t: truth values
s: indices (worlds, times, world-time pairs)
a: illocutionary acts (subtype of events)
l: utterances, locutionary acts (subtype of events)
c: contexts (specifying world and time of utterance, speaker, addressee, locutionary and illocutionary acts, etc.)

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

Sentence radicals

are specified as context-dependent propositions (Kaplanian characters).

Example; i_t, c_t are the time components of indices and contexts, i_w, c_w the world components. If not relevant, I do not specify events in semantic representations.

(8) ⟦[$_{IP}$ *John arrived*]⟧
= λcλi[$i_t < c_t \wedge i_w = c_w \wedge \exists e$[**arrive**(i)(e) ∧ AG(e,**john**)]]
= λcλi[i < c ∧ **arrive**(i)(**john**)], for short (type **cst**).

Compositional derivation; notice that all expressions are functions from contexts to functions from indices to something else.

(9) ⟦*John*⟧ = λcλi[**john**], type **cse**.

⟦*arrive*⟧ = λcλiλx[**arrive**(i)(x)], here simplified, type **cset**.

⟦*John arrive*⟧ = λcλi[⟦*arrive*⟧(c)(i)(⟦*John*⟧(c)(i)],
= λcλi[**arrive**(i)(**john**)], type **cst**.

⟦PAST⟧ = λPλcλi[i < c ∧ P(c)(i)], type **(cst)cst**.

⟦PAST [*John arrive*]⟧ = λqλcλi[i < c ∧ q(c)(i)](λcλi[**arrive**(i)(**john**)])
= λcλi[i < c ∧ **arrive**(i)(**john**)], type **cst**.

Speech Acts

Assertion of this contextualized proposition in a context c:

(10) ASSERT(c)(p) =
the speech act event in which the speaker c_s expresses at c_t the intention towards the addressee c_a that the proposition p should be accepted as part of the common ground between c_s and c_a at c_t in c_w. That is, it should be accepted that ∃i[p(i)] is true. With that, certain social sanctions apply concerning truthfulness, sufficient evidence etc.

This is a speech act event of type **a**, NOT a truth value (as it would be for Vanderveken).

Applied to our example, when the proposition is asserted at context c_0:

(11) ASSERT(c_0)(λi[i < c_0 ∧ **arrive**(i)(**john**)]), type **a**.

This is a speech act event iff in c_0 an assertion of the appropriate type is made; otherwise the term is undefined.

Speech Act Types

In linguistics, we are not concerned with particular speech acts (tokens), but rather with speech act types. A speech act type is a function from contexts into particular speech acts:

(12) λc[ASSERT(c)(λi[i < c ∧ arrive(i)(john)])], type **ca**.

This is a function that is defined for those contexts c in which the speaker asserts to the addressee that John arrived; it is undefined else. If defined, it refers to that speech act.

- "Speech act type": functions from contexts to speech acts, type **ca**.
- "Speech act sort": broader classification of speech acts and speech act types, e.g. assertions, questions, promises, requests etc. It might be possible that one speech act belongs to more than one type, e.g. question and request: *Could you open the window?*

Illocutionary Operators

(13) λqλc[ASSERT(c)(q(c))], type **(ct)ca**

Syntactic realization: Operator in Force Phrase (ForceP), cf. Rizzi (1996). It might be expressed in various ways, e.g. by declarative suffixes in Japanese or by Verb movement to second position in German.

(14) ⟦[$_{ForceP}$ A [$_{IP}$*John arrived*]]⟧
= λqλc[ASSERT(c)(q(c))](λcλi[i < c ∧ arrive(i)(john)])
= λc[ASSERT(c)(λi[i < c ∧ arrive(i)(john)])]

Other sorts of speech acts, like commands and promises, can be dealt with in similar ways, involving special illocutionay operators and sentence radicals that come with particular conditions (cf. Searle 1969). For questions, see below.

Negation

Negation can be treated as a semantic operator that is part of the sentence radical.

(15) ⟦*John didn't arrive*⟧ = λcλi¬∃i′[i′ < c ∧ $i_w = i'_w$ ∧ **arrive**(i′)(**john**)]
⟦A *John didn't arrive*⟧ = λcASSERT(c)(λi¬∃i′[i′ < c ∧ $i_w = i'_w$ ∧ **arrive**(i′)(**john**)])

Negation can also be treated as an illocutionary act:

(16) ⟦N *John arrive*⟧ = λc[NEG λi[i < c ∧ **arrive**(i)(**john**)] d

(17) NEG(c)(p) =
the speech act event in which c_S expresses at c_t the intention towards c_A that it should be accepted as part of the common ground that ¬∃i[p(i)].

The assertion of a negated proposition amounts to the same as the negation of this proposition. There might be evidence that both kinds of negations exist, e.g. German:

(18) *John ist nicht angekommen.*
John ist keineswegs angekommen.

4. Quotations and propositional attitudes

Difference between expressing speech acts and reporting speech acts

- Expressing speech acts: An event, a social act involving a locutionary act.
- Reporting speech acts: Assertion of a proposition that states that a speech act has happened.

Verbatim quotation

Relation to a linguistic expression:

(19) ⟦*John told Mary:* "*I will come*."⟧
= λcλi[i < c ∧ **tell**(i)(*I will come*)(**mary**)(**john**)]

tell expresses a relation between a speaker, an adressee and a linguistic type (the sentence *I will come*). This is to be distinguished from concrete utterances of this linguistic types, or linguistic tokens. With the simplifying assumptio that linguistic types are sets of utterances, type **lt,** the type of **tell** is then **s(lt)eet**.

(20) ⟦**tell**(i)(l)(h)(s)⟧ = 1 iff
at i, there is a context c′ with c'_S = s, c'_A = h, and c'_U is an utterance token of the linguistic type l,
provided that ⟦l⟧ is a speech act type of the sort of assertions.

The sortal restriction to speech act types of assertions is due to the meaning of *tell*; for *ask*, we would have a restriction to questions or requests.

Non-verbatim quotations

Direct speech need not be verbatim; one can report speech by quotations that would express the intention of the speakers (cf. speeches in Thukydides, *Peloponnesian War*).

Modelling by relation to a speech act type, **tell** of type **s(ca)eet**.

(21) ⟦*John told Mary:* "*I will come.*"⟧
= λcλi[i < c ∧ **tell**(i)(λc′[ASSERT(c′)(λi′[c′ < i′ ∧ **come**(i′)(c'_S)])(**mary**)(**john**)])]

The sentential argument stands for the speech act type that would be expressed by the quoted expression. The relevant interpretation of *tell* i, where @ is a variable for speech act types:

(22) ⟦**tell**(i)(@)(h)(s)⟧ = 1
iff at i, there is a context c′ with c'_S = s, c'_A = h and c'_U is of a linguistic type l
such that ⟦l⟧(c′) = @(c′), provided that @ is a speech act type of the sort of assertions.

Propositional speech report: Indirect speech

(23) ⟦*John*$_1$ *told Mary that he*$_1$ *would come*⟧
= λcλi[i < c ∧ **tell**(i)(λi′[i < i′ ∧ **come**(i′)(**john**)])(**mary**)(**john**)]

This version of **tell** is of type **s(st)eet** and can be interpreted as follows:

(24) ⟦**tell**(i)(p)(h)(s)⟧ = 1
iff at i, there is a context c′ with c'_S = s, c'_A = h and a speech act type @, and ⟦**tell**(i)(@)(h)(s)⟧ entails that s wants to add the proposition p to the common ground of c′.

Thus, propositional **tell** is reduced to speech-act related **tell**. There is no relation to the actual wording of the reported speech act, and the deictic expressions would be interpreted with respect to the original context, c.

The last version of **tell** might be type-theoretically simpler, as its sentential argument is just a proposition. However, the interpretation rule (24) does involve a speech act as well. This may explain why there are languages that appear to lack indirect speech (Kobon: Davies 1981, Matses: Fleck 2003).

Direct and indirect propositional attitudes

(25) *Mary thought that she will go.*
Mary thought: "I will go."

(26) *Mary wondered whether she would go.*
Mary wondered / asked herself: "Will I go?"

Propositional attitude reports basically express an attitude towards a proposition (or towards a partial character, to account for shifted instances of deictic advebials like *tomorrow*:

(27) λcλi[i < c ∧ **think**(i)(λi′[i_t < i'_t ∧ **go**(i)(**mary**)])]

The use of a direct quotation can be motivated if the quotation is used for a purpose that typically would put the proposition in the common ground.

(28) If α is a propositional attitude verb and @ a speech act type, then
⟦α⟧(i)(@)(x) iff there is a proposition p such that ⟦α⟧(i)(p)(x), and
i. the evidence for this is that x performed the speech act type @ compatible with α, or
ii. if x would want to inform an addressee y at i that ⟦α⟧(i)(p)(x), then this can be done by asserting ⟦α⟧(i)(@)(x).

This presupposes that propositional attitudes can be expressed linguistically, by speech acts. Not always plausible, as thought might not be fully dependent on language:

(29) a. *The fox thought that the raven was sitting in the tree.* (o.k. if foxes can think)
b. *The fox thought: "The raven is sitting in the tree."* (only o.k. if foxes can speak)

Partial Quotations

Elements of direct speech can be introduced into indirect speech ("partial quotation").

(30) a. *Maria meinte dass sie doch gar nicht gehen wolle.*
'Mary said that she MP didn't want to go.'
b. *they have said that they frankly cannot verify the peaceful nature of your program*

Proposal: The examples with indirect speech embeddings entail the existence of a speech act; this licenses the citation of expressions that were used in these speech acts.

Similar case of partial quotations in the narrow sense:

(31) Mary said that this "sucker" followed her wherever she went.

With modal particles and speech-act adverbials, using quotation marks appears infelicitous. Possible reason: They can be only interpreted with respect to the reported speech act itself, so quotation marks are unnecessary.

Speech act markers in other clauses

We find evidence for speech acts also in certain other clauses (Coniglio 2009), e.g. in the following purpose clause (where stressed *JA* marks strength of optatives).

(32) *Hans trägt einen Schlips, um JA nicht aufzufallen.*
'Hans wears a tie in order to MP not draw attention to himself.'

The use of the modal particle *JA* is justified if the speech act that expresses the purpose explicitly would contain it. Hans said to himself: *Ich will JA nicht auffallen.* 'I don't want to draw attention to myself, if at all possible'.

5. Embedded questions

Question sentence radicals

Questions are speech acts require a sentence radical that specifies a set of propositions (Hamblin 1973; Groenendijk & Stokhof 1982) or alternatively a structured proposition (von Stechow 1980). Here: Proposition set approach, for constituent and polarity questions.

(33) $[\![\textit{Who arrived?}]\!]$
$= \lambda c[\textsc{Quest}(c)(\{\lambda i[i < c \wedge \textbf{arrive}(i)(x)] \mid x\text{: }\textbf{person}\})]$

(34) $[\![\textit{Did John arrive?}]\!]$
$= \lambda c[\textsc{Quest}(c)(\{\lambda i[i<c \wedge \textbf{arrive}(i)(\textbf{john}), \lambda i\lambda\neg\exists i'[i'<c \wedge i_w=i_{w'} \wedge \textbf{arrive}(i')(\textbf{john})]\})]$

(35) $\textsc{Quest}(c)(P) =$
the speech act event in which c_S expresses at c_t the wish towards c_A to identify the proposition(s) in P that should be accepted as part of the common ground – i.e. these propositions $p \in P$ such that $\exists i[p(i)]$ should be accepted as true.

Question embedding

In most cases, the question radical is embedded; the meaning can be traced back to proposition embedding.

(36) $[\![\textit{Bill knew who arrived}]\!]$
$= \lambda c\lambda i[i < c \wedge \textbf{know}(i)(\{\lambda i[i < c \wedge \textbf{arrive}(i)(x)] \mid x\text{: }\textbf{person}\})(\textbf{bill})]$

(37) $\textbf{know}(i)(P)(x)$ iff $\forall p[p(i) \rightarrow \textbf{know}(i)(p)(x)] \wedge \forall p[\neg p(i) \rightarrow \textbf{know}(i)(\lambda i\neg p(i))(x)]$

Embedding by verbs like *wonder, ask*: Embedding of question speech act.

(38) $[\![\textit{Bill wondered who arrived}]\!]$
$= \lambda c\lambda i[i < c \wedge \textbf{wonder}(i)(\lambda c'[\textsc{Quest}(c')(\{\lambda i[i < c \wedge \textbf{arrive}(i)(x)] \mid x\text{: }\textbf{person}\})])(\textbf{bill})]$

Intuitive meaning: Bill wants to know the answer to the embedded question act type, applied to the current context; but this question act is not actually performed.

Differences: Occurrence of modal particles, possibility of quotation, root-clause syntax.

(39) *Bill *weiß / will wissen / fragt sich, wer denn kommt.*

(40) *"Did John arrive?", Bill wondered / *knew.*

6. Appositive Relative Clauses

(41) *John, who I frankly don't like, will visit me tomorrow.*

(42) *Hans, der wohl morgen kommen wird, möchte Maria treffen.*
'Hans, who (particle) will arrive tomorrow, wants to meet with Maria.'

We assume that the appositive relative clause creates a condition for the context that the speech act in question has been performed; this condition is accomodated.

Conditions as domain restrictions for functions: λx. Condition for x [Value of x]. We write $c \sim c'$ to say that contexts c, c' have the same speaker, addressee, world, time, and $c \sim i$ to say that context c and index i have the same world and time.

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

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

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

Integration of the complex DP with the appositive relative clause:

(44) $[\![\textit{will visit me}]\!] = \lambda c\lambda i\lambda x[c < i \wedge \textbf{visit}(i)(c_S)(x(c)(i))]$

$[\![[_{\text{IP}}[_{\text{DP}}\ \textit{John}\ [\textit{who}_1\ [_{\text{ForceP}}\ A\ [_{\text{CP}}\ \textit{I don't like}\ t_1]]]\ [\textit{will visit me}]]]\!]$
$= \lambda c\lambda i[[\![\textit{will visit me}]\!](c)(i)([\![[_{\text{DP}}\ \textit{John}\ [\textit{who}_1\ [_{\text{ForceP}}\ A\ [_{\text{CP}}\ \textit{I don't like}\ t_1]]]]\!](c)(i))$
$= \lambda c.\underline{\exists c'[c \sim c' \wedge \exists a[a = \textsc{Assert}(c')(\lambda i[i \sim c' \wedge \neg\ \textbf{like}(i)(x_1)(c'_S)])]}$
$\lambda i[c < i \wedge \textbf{visit}(i)(c_S)(\textbf{john})]$

The contextual presupposition of the subject DP becomes a presupposition of the clause. The resulting character is itself asserted:

(45) $[\![A\ [_{\text{IP}}\ [_{\text{DP}}\textit{John}\ [\textit{who}\ [A\ [\textit{I don't like}]]]]\ [\textit{will visit me}]]]\!]$
$= \lambda c.\underline{\exists c'[c \sim c' \wedge \exists a[a = \textsc{Assert}(c')(\lambda i[i \sim c' \wedge \neg\ \textbf{like}(i)(x_1)(c'_S)])]}$
$\textsc{Assert}(c)(\lambda i[c < i \wedge \textbf{visit}(i)(c_S)(\textbf{john})])$

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

Holler (2005) points out that 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 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.

7. Conjunctions and Quantification into Speech Acts

Conjunctions

Conjunction of speech acts is not a Boolean operation (defined for types that end in **t**).

Conjunction of speech act types $@_1$, $@_2$ is to form their functional composition:

(46) $@_1 \,\&\, @_2 = \lambda c[@_2(@_1(c))]$, type of &: **(ca)(ca)ca**

Notice that speech act conjunction may be sensitive for order:

(47) a. *Will Mary come, and when will she come?*
b. **When will Mary come, and will she come?*

But often, there is no order sensitivity, and we will have commutativity: $@_1 \,\&\, @_2 = @_2 \,\&\, @_1$

There is no disjunction of speech acts.

Quantification into speech acts

Krifka (2001) argues that the pair-list reading of questions with quantifiers is generated by quantification into question acts.

(48) *What did every guest bring?*
Intended reading: 'For every guest x: What did x bring?'
Intended answers of type: *John brought wine, Mary brought beer, Sue brought Salad.*

Observation: This reading is only available for universal quantifiers.

(49) *What did most guests bring?*
Reading: 'For what y does it hold: For most guests x, x brought y.'
Possible answer: *Most guests brought beer.*

Explanation: Speech acts can be conjoined, but not disjoined. Universal quantifiers are generalized conjunctions, hence universal quantification into speech acts is possible. Non-universal quantifiers, as e.g. *most*, also require disjunction as basic operation, hence they are not suitable for quantification over speech acts.

If A is a set of speech act types, and &A is the conjunction of all elements of A then the speech act type of (48) is as follows:

(50) $\&\{\lambda c[\textsc{quest}(c)(\lambda y \lambda \lambda i[\textbf{bring}(i)(y)(x)]) \mid x \in \textbf{guest}(i)\}$

If there are three guests, Mary, Sue and John, this amounts to the conjoined speech act *What did Mary bring, what did Sue bring, and what did John bring?*

A pair-list answer satisfies the informational need of such a question.

A case of conjunction of answers

Dotlacil (2010), among others, observed that conjunctions of DPs have a tendency towards collective interpretations. However, this vanishes if the constituents of the conjunctions are stressed; also, the preferred agreement changes.

(51) a. *John and Mary own a boat.*
b. *JOHN and MARY own(s) a boat.*

Explanation: Focus corresponds to wh-element in the question; the context question is *Who owns a boat?*; *and* is speech act conjunction.

(52) $[\![[[\textit{John and Mary}]_{\text{Focus, 1}}\ [_{\text{ForceP}}\ \text{A}\ [\text{t}_1\ \textit{owns a boat}]]]]\!]$
$= \lambda c \textsc{Assert}(c)(\lambda i[i \sim c \wedge \exists y[\textbf{boat}(i)(y) \wedge \textbf{own}(i)(y)(\textbf{john})]]) \,\&$
$\lambda c \textsc{Assert}(c)(\lambda i[i \sim c \wedge \exists y[\textbf{boat}(i)(y) \wedge \textbf{own}(i)(y)(\textbf{mary})]])$
$[\![\textit{John and Mary}]\!] = \lambda S[S(\textbf{john}) \,\&\, S(\textbf{mary})]$
$[\![[_{\text{ForceP}}\ \text{A}\ [\text{t}_1\ \textit{owns a boat}]]]\!] = \lambda x \lambda c \textsc{Assert}(c)(\lambda i[i \sim c \wedge \exists y[\textbf{boat}(i)(y) \wedge \textbf{own}(i)(y)(x)]])$

8. Conditionalized and Negated Speech Acts

Conditinalized speech acts

Speech acts can be conditionalized (relevance or biscuit conditionals; Austin 1961):

(53) If you want biscuits, there are some on the sideboard.

Analysis following Siegel (2006) involving quantification over potential speech acts.

(54) $\lambda c \lambda i[\forall i' \in R_c(i)\ [\textbf{want}(i')(\textbf{biscuits})(c_A) \rightarrow$
$\exists c'[c \sim c' \wedge c'_i = i' \wedge \textbf{occur}(i')(\textsc{Assert}(c')(\lambda i''[\textbf{there\_are}(i'')(\textbf{biscuits})])]]]$

This says that at all indices i′ that are accessible via the accessbility relatin R_c from i, if the addressee of the utterance context c wants biscuits, there is a context c′ that is similar to c (insofar speaker and addressee are the same, and possibly more) whose world is i′ and in which the assertion occurs (by the speaker, to the addressee) that there are biscuits.

Negation of speech acts

Negation is a Boolean operation, and so we do not expect that they can be applied to speech acts. However, Searle (1969) points out the difference between

(55) a. *I promise that I don't come.* b. *I don't promise that I come.*

(b), the **denegation** of a speech act, should be analyzed as a refusal to make a promise. It can be analyzed as involving a conditional speech act with an implicit *if*-part: In case you ask me that I make a promise to come, I will not do so.

(56) $\lambda c \lambda i[\forall i' \in R_c(i)\ \forall c'[c \sim c' \wedge c'_i = i' \wedge$
$\textbf{want}(i')(\lambda i'' \exists c'[i'' \sim c' \wedge \exists e[e = (\textsc{Promise}(c')(\lambda i'''[\textbf{come}(i''')(c_A)])]])]$
$\rightarrow \neg \exists c'[c' \sim c \wedge c'_i = i' \wedge \exists e[e = \textsc{Promise}(c')(\lambda i'''[\textbf{come}(i''')(c_A)])])]]$

Conditional conjunctions and disjunctions involving imperatives

(57) a. *Insult Mary and I sue you.*
b. *Stay a bit longer and I'll make you a coffee.*
c. *Go to bed now and you'll be well rested tomorrow.*

These appear to be conjunctions of an imperative followed by a threat, a promise, or another predictive speech act. However, notice that the first component need not be an imperative.

(58) Be chronically ill and the world is a desperate place.

Assumptions:

- The first sentence is not an imperative speech act, but an imperative sentence radical, type **ecst**, which have the property that the first argument is restricted to the addressee: $\lambda c \lambda i$ [**insult**(i)(**mary**)(c_A)]
- The first conjunction is interpreted as the antecedent of a conditional (Culicover & Jackendoff 1997). The prosody that makes clear that the first clause is not asserted.

Hence we have an ordinary conditional interpretation (as with biscuit conditinals):

(59) If you insult Mary again, then I sue you.

Argument for the sentence radical analysis: Modal particles do not occur.

(60) a. *Beleidige Maria doch!*
b. *Beleidige Maria *doch, und ich werde dich verklagen!*

The disjunction case

(61) Don't insult Mary or I sue you.

Meaning: 'If you insult me, then I sue you'. Typically interpreted as imperatives backed up by a threat.

Possible explanation:

- As before, the imperative clause is just a sentence radical, type **ecst**, with argument **e** restricted to addressee.
- The form [p $\vee$ q] is interpreted conditionally: if $\neg$p then q.

This results in a conditionalized speech act:

(62) If you insult Mary, I sue you.

9. Explicit Peformatives

Explicit performatives have the form of assertions, but indicate the speech act expressed by the main verb. Here: communication-oriented performatives.

(63) a. *I (hereby) promise you to come.* b. *I ask you (hereby) whether John will come.*

Proposal: Successfully asserting (a) is **the same** as promising to come. So we should allow that at one and the same context, two distinct speech acts are uttered. This can be treated as a presupposition on the context, where the proposition that is asserted is the tautology, $\top$.

(64) $\lambda c.\underline{\exists e[e = \textsc{Promise}(c)(\lambda i'[c < i' \wedge \textbf{come}(i')(c_S)])]}) \lambda i[\textsc{Assert}(c)(\top)]$

When applied to a context c, the speaker asserts a truism provided that in the same context, there is a promise by the speaker to come.

Notice that the speech act verbs, like *promise*, are not used to describe a speech act – they are dependent on a context, just like illocutionary operators. In their descriptive interpretation, they are dependent on an index:

(65) *I promised you to come.*
$\lambda c[\textsc{Assert}(c)(\lambda i[i < c \wedge \exists c'[c'_w = i_w \wedge\ c_t = i_t \wedge$
$\exists e[e = \textsc{Promise}(c')(\lambda i''[c' < i'' \wedge \textbf{come}(i'')(c_S)])]]$

10. Assertions and Concessions[1]

Difference between propositions that are asserted and propositions that are conceded for the sake of the argument, to please the other participant, etc. (cf. Merin 1999).

If we take ~ preceeding a speech act type as its denegation, grants can be analyzed as the dual of assertions. To grant Φ is to state that one would not assert $\neg\Phi$.

(66) $\lambda c \lambda s\ \textsc{Concede}(c)(\Phi) = \sim \lambda c \lambda s \textsc{Assert}(c)(\neg\Phi)$

Analysis of *at least* and *at most* as involving quantification over speech acts (not as epistemic operators, as in Geurts & Nouwen (2007).

(67) *John had at least thrée*$_F$ *martinis.*
the minimal number n such that $\exists c[\textsc{Assert}(c)(\lambda i[\text{John had n martinis at i}])] = 3$

(68) *John had at most thrée*$_F$ *martinis.*
the maximal number n such that $\exists c[\textsc{Concede}(c)(\lambda i[\text{John had n martinis at i}])] = 3$

Hence these sentences are not speech acts themselves, but **meta speech acts** indicating strategies in conversation.

This explains

- superlative morphology of *at least / at most*
- restrictions of embeddability (Nilsen 2007)

*(69) If you are *at least / more than 30 minutes late, you must report to the Registrar.*

(70) John thinks that he is at least 30 minutes late.

- Cooccurrence restrictions

(71) *John hardly drank *at least / more than three martinis.*

hardly marks a concession or weak assertion; this is not compatible with *at least*.

11. Conclusion

There are many cases in which the concept of embedded speech acts can be fruitfully applied.

These cases do not force us to assume an analysis of speech acts as propositions.

Chances are good that a model theory can be developed in which speech acts are regular semantic objects, explaining their embeddability.

[1] Joint ongoing work with Arik Cohen.