CHAPTER 13

COMPOSITIONALITY IN DISCOURSE FROM A LOGICAL PERSPECTIVE

TIM FERNANDO

13.1 Introduction

As a piece of language that can span several sentences, discourse goes beyond the expressions to which the Principle of Compositionality (PC) is commonly applied.

(PC) The meaning of a complex expression *e* is determined by the meaning of its parts and how those parts are put together to form *e*.

The Principle of Compositionality is often contrasted with the Context Principle (X) of Frege (e.g. Janssen, 2001).

(X) The meaning of a word *w* can only be given in the context of a sentence where *w* appears.

We can relate (X) roughly to (PC) if we take a complex expression to be a sentence, and its parts to be the words in it. Turning to discourse, we shall see that we must step beyond sentences in isolation, and that as we do, various notions of meaning break down under the weight of compositionality (PC). To understand how and why this should be the case, we shall appeal to a perspective on (PC) that brings (PC) in line with a broad construal (X′) of (X).

(X′) The meaning of an expression *e* can only be given in the context of a discourse where *e* appears.

The passage from (PC) to (X′) is, in a nutshell, what the present chapter is about, its bones. We flesh out these bones with material scattered in the *Handbook of Logic and*

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 280 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

Language (van Benthem and ter Meulen, 1997), the obvious reference for the logical perspective on natural language,[1] and with further empirical considerations.

13.1.1 Scope

Let us start with what we take the scope of compositionality in discourse to be in the present chapter. (PC) describes an interpretation with 'a complex expression' fed as input, and 'meaning' returned as output. Discourse aside, bounds are, in practice, imposed on these PC-inputs and PC-outputs. For instance, in Montague Grammar,

(a1) PC-inputs are drawn from a 'disambiguated language' (with ambiguities that plague natural language resolved away)

and

(a2) PC-outputs are confined to truth-conditional aspects (to be supplemented, for example, by Gricean implicatures).

Both assumptions (a1) and (a2) abstract away important features of natural language—ambiguity in the case of (a1), and non-literal meaning in the case of (a2). But as neither assumption is forced by (PC), either can be challenged when applying (PC) to discourse.

With respect to (a2), there is a well-known minimal pair attributed to Barbara Partee that is generally employed as an argument that truth conditional meaning is too coarse for pronoun reference.

(13.1) a. I dropped ten marbles and found all of them, except for one. It is probably under the sofa.
b. I dropped ten marbles and found only nine of them. #It is probably under the sofa.

The first sentences in (13.1a) and (13.1b) are truth-conditionally equivalent, but as two-sentence discourses, (13.1a) and (13.1b) differ. One way to account for this difference is to recognize (13.1a) as a legal PC-input, but not (13.1b). (This way we need not worry further about (PC).) Otherwise, if (13.1a) and (13.1b) are both subject to compositional interpretation, then their difference in meaning must (by (PC)) be traced to a difference in the meanings of their parts or a difference in the way these parts are put together to form (13.1a) and (13.1b). Some leeway is available here because PC-inputs may go beyond strings of English words and punctuation symbols displayed in (13.1). Under (a1), for instance, the PC-input for (13.2) must fix what *can be fun*: relatives who are visiting, or visits to relatives.

(13.2) Visiting relatives can be fun.

[1] No less than five chapters in that Handbook are directly relevant to compositionality in discourse—namely, those written by (i) Partee with Hendriks, (ii) van Eijck and Kamp, (iii) Janssen, (iv) Muskens, van Benthem, and Visser, and (v) Beaver.

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 281 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

Of course, explaining the difference observed in (13.1) through the structure of PC-inputs need not commit us to preserving a truth-conditional picture (a2) of meaning. It is natural to expect more complicated outputs from more complicated inputs. Indeed, what better motivation can there be to complicating the inputs than that we demand more from the outputs? That said, we may demand more from the outputs precisely to avoid complicating the inputs—whence the tendency to construe (13.1) as an invitation to step beyond truth-conditions.

It is a purely technical question whether or not to count (13.1a) but not (13.1b) among the allowed PC-inputs. The more substantive problem is the possibility of truth-value gaps. This possibility looms over sentences such as (13.3), if not (13.4).

(13.3) He beats his donkey.

(13.4) If Pedro owns a donkey, he beats his donkey.

To assign a truth-value to (13.3), we must specify who *he* and *his donkey* are. Under an anaphoric reading of (13.4), *he* is *Pedro*, and *his donkey* is *a donkey Pedro owns*. Analyses of such a reading and also of (13.1) and (13.3) have been developed that expand conceptions of meaning from truth-conditions to context change. These are reviewed in Section 13.2 below, where context change is shown to support (among other things) the assignment of truth-values to suitably disambiguated inputs.

These disambiguated inputs, as in (a1), take anaphor(a) resolution for granted. A classic illustration of the complexity of anaphor resolution is the minimal pair (13.5) due to Winograd (1972).

(13.5) a. The authorities denied the marchers a permit because they feared violence.
b. The authorities denied the marchers a permit because they advocated violence.

Common sense suggests that in (13.5a) *they* are *the authorities* whereas in (13.5b) *they* are *the marchers*. Aside from the piece of 'world knowledge' that between authorities and marchers, the former can be expected to fear violence and the latter to advocate it, there is our understanding of the word *because*, which in a sentence of the form *S because S′* offers *S′* as an explanation of *S*. In practice, the word *because* is at times dropped and (13.6a) understood as (13.6b), with the danger that (13.6a) could instead be read as say, (13.6c).

(13.6) a. *S.S′*.
b. *S* because *S′*.
c. *S* and then *S′*.

An example analysed at length in Asher and Lascarides (2003) is (13.7).

(13.7) a. Max fell. Mary pushed him.
b. Max fell because Mary pushed him.

The tendency to read (13.7a) as (13.7b) reflects a certain familiarity with stories of pushes leading to falls. If we change the second sentence *S′* as in (13.8), then (13.6a) is more likely to be read as (13.6c).

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 282 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

(13.8) Max fell. Mary helped him get up.

Jumping from (13.6a) to (13.6b) is a leap that we should be prepared to reconsider, as illustrated by (13.9).

(13.9) Max fell. Mary pushed him. Lying on the ground, he was unable to resist her push.

Why not avoid such leaps by restricting interpretation in (PC) to that which is explicit?

Because (very briefly) a discourse is seldom, if ever, fully explicit and PC-inputs generally go beyond the surface forms that are explicitly given. By mentioning 'parts' and 'how those parts are put together to form *e*', (PC) raises the possibility of non-explicit parts and of any number of ways to put those parts together. At one extreme, we note that parts are all explicit and can only be put together one way in

Example PL: Discourse as a theory in predicate logic A discourse here is a set of sentences, where a sentence φ is evaluated against a suitable model M by a satisfaction relation $\models$. Relative to a set $\mathcal{M}$ of such models, the meaning $[\![D]\!]_{\mathcal{M}}$ of a discourse D is defined to be the set of models in $\mathcal{M}$ that $\models$-satisfy every sentence in D

$$[\![D]\!]_{\mathcal{M}} = \{M \in \mathcal{M} \mid (\forall \varphi \in D)\ M \models \varphi\}$$

so that compositionality in discourse comes down to the equation

$$[\![D]\!]_{\mathcal{M}} = \bigcap_{\varphi \in D} [\![\{\varphi\}]\!]_{\mathcal{M}}\ .$$

Example PL misrepresents natural language discourse in at least three ways. First of all, for pronouns and many other 'holes' filled by context, it is useful to form a discourse from formulas with free variables, pairing a model M with a variable assignment f in M before applying $\models$ to the formula φ. In connection with compositionality, it is worth noting that f constitutes a notion of context that is allowed to vary when defining

$$(M, f) \models \varphi$$

by induction on a formula φ. Secondly, if, for instance, the free variables are to be interpreted appropriately, then simply throwing the parts together in a set will not do for compositional interpretation. At the very least, the order of English sentences in a text is significant. (Consider just about any two-sentence English text where the first sentence introduces an entity referred to by a pronoun in the second sentence.) In addition, there is the possibility of putting the parts together using words such as *because* and *and then*. This brings us to the third problem with Example PL as a model of discourse. The parts of a natural language discourse should not be limited to sentences but must range also over subsentential units such as connectives between sentences (including perhaps *if . . . then . . .*, which figures in the donkey sentence (13.4) above). These three problems taken together suggest that discourse processing is far more than interpreting a discourse compositionally against pairs (M, f) of models M and variable assignments f in M.

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 283 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

A good deal of work must go to fashioning discourse into a structure that can serve as a PC-input. These structures are called *Discourse Representation Structures* (DRSs) in *Discourse Representation Theory* (DRT, Kamp and Reyle, 1993). The construction of DRSs from natural language text is distinguished from the model-theoretic interpretation of DRSs (the latter being what interpretation in (PC) is traditionally identified with). It is noteworthy that early on, Kamp should write that

> it seems to me that the rules for the construction of discourse representations have at least as good a claim to being constitutive of meaning as the clauses that make up the definition of truth. (Kamp, 1979: 409)

and that after about a decade of work, he should declare

> the principal challenge to DRT, experience indicates, is the exact formulation of the construction algorithm. (Kamp, 1990: 37)

The challenge in formulating the construction algorithm has to do in no small measure with semantic reasoning within the algorithm.[2] Take, for example, the simple variant (13.10) of (13.4).

(13.10) If Pedro owns a donkey, he beats the quadruped.

We can link *the quadruped* to *a donkey Pedro owns* if we can infer the latter is a quadruped. In (13.7), repeated below, a straightforward deduction (or induction) will not do to leap from (13.7a) to (13.7b).

(13.7) a. Max fell. Mary pushed him.
b. Max fell because Mary pushed him.

For any fall, we can expect many more pushes to temporally precede it than cause it (to say nothing of pushes after the fall). The plausibility of (13.7b), given (13.7a), depends on the manner in which (13.7a) reports the push immediately after the fall. We must draw on knowledge not only about the world of pushes and falls but also about how language is used to describe that world. A common view concerning language use is that the sentences in a discourse must relate to each other so as to form a coherent whole (Halliday and Hasan, 1976; Hobbs, 1979; Polanyi, 1985; Grosz and Sidner, 1986; Mann and Thompson, 1988; Asher and Lascarides, 2003; Webber, Stone, Joshi, and Knott, 2003; Kehler, 2002). The concrete issue for a two-sentence discourse is how to present it as a PC-input: as (13.6a) or as some complex such as, for example, (13.6b) or (13.6c) that specifies how S is related to S'?

(13.6) a. $S.S'$.
b. S because S'.
c. S and then S'.

[2] By contrast, in ordinary predicate logic, the formation of predicate logic formulas depends in *no* way on the subsequent semantic manipulations of these formulas (in accordance with their model-theoretic interpretation). The crucial difference is that the DRS construction algorithm starts from natural language input with much information necessary for interpretation left implicit.

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 284 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

An instantiation of the sequence (13.6a) can be described as incoherent insofar as it is unclear how S is related to S' in it. Consider

(13.11) Pat bought a hamster in 1996. Three is a prime number.

There is no denying the oddness of (13.11). But is it un-interpretable? The two sentences are separately interpretable, and we might hope that a context can be found for combining them coherently. One attempt (albeit feeble) is

(13.12) Pat bought a hamster in 1996. Three is a prime number. These are the first two facts about Pat and three which spring to mind.

The last sentence in (13.12) presupposes a topic (viz., *the first two facts about Pat and three which spring to mind*) that an interpreter is asked to take for granted. Two questions arise. First, can we, in general, render two otherwise disconnected sentences into a coherent discourse by forming a topic around the conjunction of their subjects? And second, can the incremental interpretation of a coherent discourse involve the interpretation of parts that are not coherent (when not considered in full; e.g. (13.11))? An affirmative answer to the first question would appear to neuter the constraints on discourse that a notion of coherence is designed to impose. An affirmative answer to the second would imply that we should not require every PC-input to be coherent, even if we require the whole combination at the end to be coherent. As tempting as it may be to subject discourse structure to scrutiny under compositionality (PC), we need not insist that all that structure be present in a PC-input or even in a PC-output. Some of that structure might emerge only through further processing. We are, after all, free to apply (PC) to any of a number of processes for interpreting discourse; these processes diverge, to varying degrees, from assumptions (a1) and (a2).

(a1) PC-inputs are drawn from a disambiguated language.
(a2) PC-outputs are confined to truth-conditional aspects.

In place of these assumptions, we explore two below for discourse

(b1) PC-inputs have enough structure to associate truth-conditions with them

and

(b2) PC-outputs feed into the interpretation of further PC-inputs.

Assumptions (b1) and (b2) are somewhat modest departures from (a1) and (a2), compared to the contention from Ginzburg and Cooper (2004: 297) that for dialogue, 'the updates resulting from utterances cannot be defined in purely semantic terms', destroying the Montagovian picture of compositionality as a homomorphism from a syntactic to a semantic algebra. We will return to Ginzburg and Cooper briefly below (in Section 13.4.2). Until then, we will concern ourselves with challenges of the sort illustrated above, arising in discourse that, short of dialogue, consists of declarative sentences.

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 285 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

13.1.2 Outline

Behind assumptions (b1) and (b2) is the truism that there is more to discourse than the succession of sounds we hear (or symbols we read), making discourse interpretation all about what structures we associate with these sounds. Compositionality (PC) is a constraint on transformations of these structures that yield what can be called (perhaps rather too loosely) meanings. There is a tradition in formal semantics of basing compositionally derived meanings on a 'reality' external to the heads of language users, employing model-assignment pairs (M, f) for this purpose (Jackendoff, 1996; Hamm, Kamp, and Lambalger, 2006). In Section 13.2, we proceed from such pairs (M, f) to a conception of PC-outputs that captures both external and internal aspects of meaning. That conception is closely tied to assumption (b2), and depends on a notion of context against which to understand PC-inputs and PC-outputs. Contexts and PC-outputs are reformulated proof-theoretically in Section 13.3, construing formulas as types (e.g. Troelstra and Schwichtenberg, 2000).

In Section 13.4, we look more closely at PC-inputs and turn to assumption (b1). Whereas Section 13.2 starts from model-assignment pairs (M, f), Section 13.4 revolves around representations that express in PC-inputs both the explicit (verbal) and implicit (non-verbal) components of discourse. (Representing enough implicit information to interpret what is made explicit is arguably the main challenge in discourse processing.) The implications of these representations for meanings are considered, including a notion of *part* distinct from (but related to) that mentioned in (PC). If there is a tendency to view PC-outputs as pertaining to a reality outside the mind, there is equally a temptation to regard PC-inputs as representations that somehow record internal cognitive processes.

13.2 Meaning and context

This chapter would not be written from a logical perspective if it did not (to put it very mildly) link meaning to truth. The standard picture of truth in logic is that of satisfaction $\models$ of a well-formed formula φ relative to a model M and a variable assignment f in M. For example, we may associate with (13.13) the formula $admire(x_{spk}, p, x_{now})$ built from variables x_{spk} and x_{now} (for the speaker and speech time) and constant p (for *Pat*), and interpreted according to (13.14).

(13.13) I admire Pat.

(13.14) $(M, f) \models admire(x_{spk}, p, x_{now})$ iff $admire^M(f(x_{spk}), p^M, f(x_{now}))$

Inspecting (13.14), one may ask: why separate out the variable assignment f from M? It is a trivial matter to eliminate talk of variable assignments by

1. reconstruing free variables (which we can assume never occur bound, renaming bound variables if necessary) as fresh constants;

and then

2. reconceptualizing (M,f) as a model over an enlarged vocabulary with additional constants.

In the case of (13.14), we reconstrue x_{spk} and x_{now} as fresh constants s and n, to be interpreted over an expansion M_f of M with $s^{M_f} = f(x_{spk})$ and $n^{M_f} = f(x_{now})$ so that

$$M_f \models admire(s,p,n) \text{ iff } admire^{M_f}(s^{M_f}, p^{M_f}, n^{M_f})$$
$$\text{iff } (M,f) \models admire(x_{spk}, p, x_{now}).$$

For natural language applications, one reason not to bury f in a model (over an enlarged vocabulary) comes from so-called two-dimensional semantics (Kaplan, 1975). We discuss this below before reformulating all constants as variables, resulting in pairs (M,f) where the vocabulary of M has been stripped of all constants and the domain of f has been expanded to record the interpretations of these constants. This elimination of constants paves the way for a notion of context against which to understand the assumption (b2) that PC-outputs feed into PC-inputs. PC-inputs are constructed throughout this section from formulas φ, while PC-outputs are derived (one way or another) from pairs (M,f).

13.2.1 Variable assignments as contexts

As every speaker of English knows, the claim made by a sentence such as (13.13) depends on who utters it and when. Change the circumstances of its utterance and the claim changes. A way to capture these systematic variations is to associate variables with pertinent aspects of the utterance such as the speaker and the speech time, and formulate an utterance as a variable assignment f that specifies speaker $f(x_{spk})$, speech time $f(x_{now})$, etc. We can then bring out the dependence of the meaning of (13.13) on its utterance by presenting (13.13) as the formula $admire(x_{spk}, p, x_{now})$ (rather than $admire(s,p,n)$). Next, let us fix some set $\mathcal{M}$ of models, each of which we regard as a *possible world*. Then an utterance f of a formula φ picks out a set of possible worlds—or in possible worlds semantics, a *proposition*. That is, borrowing terminology from Kaplan (1979), the $\mathcal{M}$-*content* of φ at f is the proposition

$$\mathsf{content}_{\mathcal{M}}(\varphi,f) = \{M \in \mathcal{M} \mid (M,f) \models \varphi\}$$

consisting of models in $\mathcal{M}$ that, paired with f, satisfy φ. Abstracting over the utterance f, the function mapping f to $\mathsf{content}_{\mathcal{M}}(\varphi,f)$ is the $\mathcal{M}$-*character* of φ

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 287 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

$\mathsf{character}_{\mathcal{M}}(\varphi) = (\lambda f \in F_{\mathcal{M}})\ \mathsf{content}_{\mathcal{M}}(\varphi, f)$

where $F_{\mathcal{M}}$ is the set of variable assignments in models from $\mathcal{M}$. A member f of $F_{\mathcal{M}}$ is called an $\mathcal{M}$-*context*, and is understood to be fixed when speaking of $\mathcal{M}$-content. We let f vary over $F_{\mathcal{M}}$ to reveal the $\mathcal{M}$-character of a formula. Contextual factors such as the speaker and speech time (in f) are not strictly part of *what* content is (understood as a subset of $\mathcal{M}$) although they determine *how* that content is expressed by constituting an input f to the character of a formula.

Turning now to compositionality (PC), we have two obvious candidates for the meaning of an expression φ, namely $\mathsf{content}_{\mathcal{M}}(\varphi, f)$ and $\mathsf{character}_{\mathcal{M}}(\varphi)$. These notions are progressive abstractions over the Fregean idea that a sentence refers to one of two truth-values. These truth-values are relativized to possible worlds by content, which amounts to the Carnap–Montague conception of intension as a function mapping a possible world to reference

φ is true at M iff $M \in \mathsf{content}_{\mathcal{M}}(\varphi, f)$.

Character recognizes a further layer, namely that of utterances. The step from reference to content and the step from content to character both allow variations in a parameter previously held fixed; a possible world M is fixed for reference, while an utterance f is fixed for content. As we will see next, however, the problem is that λ-abstracting over utterances and possible worlds does not on its own account for certain changes observed in interpreting discourse.

13.2.2 Context change

A crucial component of context that is especially obvious when processing a discourse consisting of more than one sentence is the surrounding text, also known as co-text. Co-text can plug truth-value gaps associated with presupposition failure. Consider

(13.15) a. Pat's car is yellow.
b. Pat has a car. It is yellow.
c. Pat has a car. Pat's car is yellow.
d. If Pat has a car, it/Pat's car is yellow.

Neither (13.15a) nor the negation of (13.15a) can be said to be true if Pat has no car. This possibility is ruled out in (13.15b) and (13.15c) and side-stepped in (13.15d). The idea, roughly, is that an indefinite description (such as *a car*) introduces a *discourse referent* (Karttunen, 1976) that a pronoun (such as *it*) or a definite description (such as *the car*) can subsequently pick up.

In DRT,[3] discourse referents are treated as variables that may or may not be in the domain of a variable assignment f. That is, a variable assignment f is understood to specify not only contextual factors such as the speaker and speech time, but also

[3] A cousin of DRT with about the same age and features is *File Change Semantics* (Heim, 1982).

'familiar' discourse referents that have been previously introduced in the discourse. A useful annotation on texts is to hang variables as superscripts when they are introduced (in which case they are said to be 'novel'), and to attach them as subscripts where they are required to be familiar.

(13.16) a. (Pat's car)$_x$ is yellow.
b. Pat has (a car)^x. It$_x$ is yellow.
c. Pat has (a car)^x. (Pat's car)$_x$ is yellow.
d. If Pat has (a car)^x, it$_x$/(Pat's car)$_x$ is yellow.

The upshot is that a sentence can add discourse referents and/or conditions on discourse referents. Accordingly, a PC-input in DRT, called a DRS (for Discourse Representation Structure), is taken to be a pair (U, C) of a set U of discourse referents, and a set C of conditions. The old picture of satisfaction $\models$ applies to conditions (in C), but a DRS as a whole can be conceived, relative to a model M, as a programme with inputs and outputs that range over variable assignments in M. A DRS (U, C) that is fed a variable assignment f in M as input can return a variable assignment g in M as output if g extends f to the discourse referents in U and (M, g) satisfies every condition in C. More precisely, we encode the input/output behaviour of (U, C) in M as a binary relation $[\![(U, C)]\!]_M$ between variable assignments f and g in M such that

$$f[\![(U,C)]\!]_M g \text{ iff } f \subseteq g \text{ and } \mathit{domain}(g) = \mathit{domain}(f) \cup U \text{ and for each } \varphi \in C,\ (M,g) \models \varphi.$$

A condition φ can be turned into the DRS $(\emptyset, \{\varphi\})$, from which the $\mathcal{M}$-content of φ can be extracted

$$M \in \mathsf{content}_{\mathcal{M}}(\varphi, f) \text{ iff } (M,f) \models \varphi \text{ iff } f[\![(\emptyset, \{\varphi\})]\!]_M f$$

for $M \in \mathcal{M}$. For any DRS K, we assume *the negation* $\neg K$ *of* K is a condition that is satisfied by (M, f) precisely if f falls outside the domain of $[\![K]\!]_M$

$(M, f) \models \neg K$ iff there is no g such that $f[\![K]\!]_M g$.

The relations $[\![K]\!]_M$ are not unlike the input/output interpretations of programmes in *Quantified Dynamic Logic* (Harel, 1984), with $\neg K$ amounting to the dynamic logic formula $[K]\bot$ saying that some contradictory formula $\bot$ holds whenever K terminates (which is to say: K cannot terminate). Starting with Barwise (1987), various systems of 'dynamic semantics' mixing DRT and dynamic logic have been proposed, such as *Dynamic Predicate Logic* (DPL, Groenendijk and Stokhof, 1991). An example commonly mentioned as motivation for such systems is Geach's 'donkey sentence' (13.17).

(13.17) If (a farmer)^x owns (a donkey)^y he$_x$ beats it$_y$.

To analyse a conditional such as (13.17), DRT builds a condition $K \Rightarrow K'$ from DRSs K and K', which is satisfied by a pair (M, f) exactly if the domain of $[\![K']\!]_M$ includes every $[\![K]\!]_M$-output on input f

$(M, f) \models K \Rightarrow K'$ iff $(\forall g$ such that $f[\![K]\!]_M g)(\exists h)\ g[\![K']\!]_M h$.

Example (13.17) is then translated as the DRS condition

$(\{x, y\}, \{\text{farmer}(x), \text{donkey}(y), \text{own}(x, y)\}) \Rightarrow (\emptyset, \{\text{beat}(x, y)\})$.

We can also define $\Rightarrow$ in terms of negation $\neg$, treating $(U, C) \Rightarrow K'$ as an abbreviation of $\neg(U, C \cup \{\neg K'\})$. In DPL, this reduction can be put quite perspicuously as the equivalence between the material conditional $\varphi \Rightarrow \psi$ and $\neg(\varphi \wedge \neg\psi)$, familiar from Boolean logic. The crucial difference is that in dynamic semantics, a formula φ is interpreted as an input/output relation $r(\varphi)$ with $\wedge$ expressing relational composition $\circ$

$$\begin{aligned} r(\varphi \wedge \psi) &= r(\varphi) \circ r(\psi) \\ &= \{(a, b) \mid (\exists c)\ a\ r(\varphi)\ c \text{ and } c\ r(\psi)\ b\}\ . \end{aligned}$$

In DRT, the obvious way to conjoin two DRSs (U, C) and (U', C') is through the binary operation *merge* $\uplus$ that unions together the respective components

$(U, C) \uplus (U', C') = (U \cup U', C \cup C')$.

For example,

$(U, C) = (U, \emptyset) \uplus (\emptyset, C)$

and for any model M,

$[\![(U, C)]\!]_M = [\![(U, \emptyset)]\!]_M \circ [\![(\emptyset, C)]\!]_M$.

A case in which

$[\![K \uplus K']\!]_M \neq [\![K]\!]_M \circ [\![K']\!]_M$

is given by $K = (\emptyset, \{x = x\})$ and $K' = (\{x\}, \emptyset)$. To rule out such examples, let us define (U', C') to be (U, C)-*novel* if *no* discourse referent in U' belongs to U or appears in a condition in C. Clearly, if K' is K-novel, then

$[\![K \uplus K']\!]_M = [\![K]\!]_M \circ [\![K']\!]_M$.

Notice that the input/output interpretation $[\![K]\!]_M$ of a DRS K depends on the choice of a model M. We can abstract over such choices by collecting possibilities (M, f) as follows. An *information state* is a set σ of pairs (M, f) of models M and variable assignments f in M. A DRS K can then be interpreted as a function K-update on information states that maps σ to the information state

$$K\text{-update}(\sigma) = \bigcup_{(M,f)\in\sigma} \{(M, g) \mid f[\![K]\!]_M g\}$$

gathering the $[\![K]\!]_M$-outputs on input f, for (M,f) ranging over σ. The intuition behind an information state σ is that each (M,f) in σ is a possibility, and that an information state σ' carries at least as much information as σ, notated $\sigma \sqsubseteq \sigma'$, precisely if every possibility in σ' fleshes out one in σ

$\sigma \sqsubseteq \sigma'$ iff $(\forall (M,f') \in \sigma')(\exists f \subseteq f')\ (M,f) \in \sigma$

(where (M',f') *fleshes out* (M,f) if $M = M'$ and $f' \supseteq f$.) The two components U and C in a DRS $K = (U,C)$ represent two different ways of adding information—by expanding the domain of a variable assignment (to include a discourse referent in U), and by eliminating a possibility (if it fails to satisfy a condition in C). K increases the information content in an information state σ in that

$\sigma \sqsubseteq K\texttt{-update}(\sigma)$

because

whenever $f[\![K]\!]_M g, f \subseteq g$

for all M. By contrast, information need not increase in systems of dynamic semantics (such as DPL) that use random assignment $x :=?$ (in place of the DRS $(\{x\},\emptyset)$) with a relational interpretation $r(x :=?)$ given by

$$f\ r(x :=?)\ g \ \text{ iff } \ \mathit{domain}(g) = \mathit{domain}(f) \cup \{x\} \text{ and}$$
$$(\forall y \in \mathit{domain}(g) - \{x\})\, f(y) = g(y)$$

that does not require $f \subseteq g$. In practice, however, it is unclear that a random assignment $x :=?$ is ever applied to a variable x that has already been assigned a value. For instance, when interpreting a pair of DRSs K and K' in sequence (representing say, a two-sentence text), it is natural to assume K' is K-novel. For the record,

Proposition 1. *Given DRSs K and K′, if K′ is K-novel, then*

$[\![K \uplus K']\!]_M = [\![K]\!]_M \circ [\![K']\!]_M$

for any model M, and so

$(K \uplus K')\texttt{-update}(\sigma) = K'\texttt{-update}(K\texttt{-update}(\sigma))$

for any information state σ.

The significance of Proposition 1 lies in its relevance to the basic intuition behind DRT: a discourse consisting of a sequence $S_1 \ldots S_n$ of sentences is processed one sentence at a time from left to right, by constructing (for $1 \leq i \leq n$) a DRS K_i representing sentence S_i in the context of representations $K_1 \cdots K_{i-1}$ of the preceding sequence $S_1 \ldots S_{i-1}$. For the discourse referents in K_i to be novel, we require that K_i be $(K_1 \uplus K_2 \uplus \cdots \uplus K_{i-1})$-novel. Let us call $K_1 K_2 \cdots K_n$ an *admissible DRS sequence* if this holds for all $i > 1$ and $\leq n$. Now, it is immediate from Proposition 1 that an admissible DRS sequence $K_1 \cdots K_n$ can be $\uplus$-merged for sequential interpretation against a model M

$[\![K_1 \uplus K_2 \uplus \cdots \uplus K_n]\!]_M = [\![K_1]\!]_M \circ [\![K_2]\!]_M \cdots \circ [\![K_n]\!]_M$

or against information states σ

$(K_1 \uplus \cdots \uplus K_n)\text{-update}(\sigma) = K_n\text{-update}(\cdots K_1\text{-update}(\sigma) \cdots)$.

Evidently, compositionality (PC) holds for a complex expression given by an admissible DRS sequence $K_1 \cdots K_n$ with parts K_i (for $1 \leq i \leq n$), and meanings

$[\![K_1 \uplus \cdots \uplus K_n]\!]_M$ and $[\![K_i]\!]_M$

(if we can single out a model M) or (failing that) meanings

$(K_1 \uplus \cdots \uplus K_n)$-update and K_i-update

(Table 13.1). Each part K_i can, in turn, be conceived as a complex expression subject to compositionality, under the clauses for $\models$ and $[\![\cdot]\!]_M$. Compositionality in this case, however, pertains not so much to discourse-as-a-sequence-of-sentences, but rather to subsentential structure.

By the associativity of $\uplus$ and $\circ$, we can parenthesize $K_1K_2K_3$ either as $(K_1K_2)K_3$ (in accordance with the aforementioned intuition behind DRT) with parts K_1K_2 and K_3, or as $K_1(K_2K_3)$ with parts K_1 and K_2K_3. Implicit in the choice of parts K_i in Table 13.1 is the use of a multi-ary operation (of multiple arities $n \geq 1$) that can be derived from iterations of a binary operation under any system of parenthesization. Indeed, we might extend DRSs with a sequential connective; interpreted as relational composition $\circ$

$[\![K; K']\!]_M = [\![K]\!]_M \circ [\![K']\!]_M$

but at the cost of losing the decomposition of *every* DRS to a pair (U, C) of sets of discourse referents and conditions. Our restriction to admissible sequences KK' (where K' is K-novel) allows us to retain the uniform picture of a DRS as a pair (U, C) by flattening $K_1; K_2; \cdots; K_n$ to $K_1 \uplus K_2 \uplus \cdots \uplus K_n$.

The novelty requirement in admissible DRS sequences captures only part of what assumption (b2) is about.

(b2) PC-outputs feed into the interpretation of further PC-inputs.

Complementing novelty is familiarity, which for a sequence $(U, C)(U', C')$ of DRSs and an initial DRS (U_0, C_0) comes to the clause

(†) every discourse referent appearing in C' but not in U' is in U
and every discourse referent appearing in C_0 is in U_0.

Table 13.1 C-inputs and C-outputs in § 2.2

C-input	admissible DRS sequence $K_1 \cdots K_n$ with parts K_i
C-output	$[\![K_1 \uplus \cdots \uplus K_n]\!]_M$ (relative to a fixed model M) $(K_1 \uplus \cdots \uplus K_n)$-update (collecting different M's)

To make (†) concrete, let us return to (13.16a), (13.16b), and (13.16c), repeated below.

(13.16) a. $(\text{Pat's car})_x$ is yellow.
b. Pat has $(\text{a car})^x$. It_x is yellow.
c. Pat has $(\text{a car})^x$. $(\text{Pat's car})_x$ is yellow.

Let us assign (13.16b) the admissible DRS sequence (U, C) $(\emptyset, \{\text{yellow}(x)\})$ where $x \in U$. (†) then holds in (13.16b) but not in (13.16a) or in (13.18).

(13.18) Not all cars have dull colours. $(\text{Pat's car})_x$ is yellow.

As the first sentence in (13.18) fails to *make x* familiar, (13.18) can only be interpreted in a context where x is already familiar. Both (13.16a) and (13.18) *presuppose* an x that is *Pat's car* (Beaver, 1997). By contrast, both (13.16b) and (13.16c) assert *Pat has a car x*, which we might assume qualifies x to be *Pat's car*. Between (13.16a)/(13.18) and (13.16b)/(13.16c) are discourses such as (13.19) where a discourse referent y is assumed familiar that intuitively can be linked but not equated with a familiar discourse referent x (Clark, 1975; Asher and Lascarides, 2003).

(13.19) Pat has $(\text{a car})^x$. $(\text{The engine})_y$ needs repair.

In general, satisfying definite descriptions can be a difficult task involving implicit information from any number of sources, including the visual scene. Often presuppositions must simply be accommodated (Lewis, 1979), such as that represented by the proper name *Pat* in (13.16), (13.18) and (13.19). In DRT, *Pat* is treated as a unary relation, rather than a constant, leading to the translation (13.20) of the first sentence of (13.16b), (13.16c) and (13.19).

(13.20) Pat_u has $(\text{a car})^x$.
$(\{u, x\},\ \{\text{pat}(u), \text{car}(x), \text{has}(u, x)\})$

In (13.20), the presupposed discourse referent u (appearing as a subscript) is included in the DRS's set of discourse referents (alongside the variable x in superscript). A wide-ranging account of presuppositions is given in van der Sandt (1992) that treats presupposition triggers as anaphoric. Inasmuch as anaphora resolution lies outside compositional interpretation, however, this puts presuppositions beyond the scope of compositional semantics.

13.3 A PROOF-THEORETIC PERSPECTIVE

The crucial notion distinguishing a DRS K from an ordinary formula φ is that of a discourse referent, analysed in the step from $\mathsf{content}_{\mathcal{M}}(\varphi, f)$ and $\mathsf{character}_{\mathcal{M}}(\varphi)$ in Section 13.2.1 to the meanings $[\![K]\!]_M$ and K-update in Section 13.2.2. Instead of moving to DRSs, we can extract discourse referents from more or less ordinary formulas φ if we trade in model-theoretic semantics (centered around $\models$) for a proof-theoretic perspective (involving types).

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 293 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

13.3.1 Dependent types, t-contexts, and records

The idea of interpreting a formula φ as a type $[\varphi]$ is that

(i) the type $[\varphi]$ consists of 'constructive' proofs of φ

where

(ii) a type is defined relative to a finite sequence of variables paired with types

as in Martin-Löf type theory (Sundholm, 1986 Ranta, 1994).

Let us explain (i) and (ii) in turn.

The type constructs we require for (i) introduce dependencies in Cartesian products $A \times B$ and function spaces $A \to B$. Recall that $A \times B$ consists of pairs $\langle a, b\rangle$ of members a of A and b of B

$$A \times B = \{\langle a, b\rangle \mid a \in A \text{ and } b \in B\}$$

while $A \to B$ consists of functions mapping members a of A to members b of B

$$A \to B = \{f \mid f \text{ maps } a \in A \text{ to } b \in B\}.$$

Now, for dependencies, we allow the second type B to depend on an object of A, writing $B_{x/a}$ for the type obtained from B by instantiating x as a. We can then form the type $(\Sigma x \in A)B$ of pairs $\langle a, b\rangle$ consisting of members a of A and b of $B_{x/a}$

$$(\Sigma x \in A)B = \{\langle a, b\rangle \mid a \in A \text{ and } b \in B_{x/a}\}$$

as well as the type $(\Pi x \in A)B$ of functions mapping members a of A to members of $B_{x/a}$

$$(\Pi x \in A)B = \{f \mid f \text{ maps } a \in A \text{ to } b \in B_{x/a}\}.$$

We can express universal quantification through Π, and existential quantification through Σ, as illustrated in (13.21) and (13.22), respectively.

(13.21) Every ant bites.
$(\Pi x \in \text{ant})\ \text{bites}(x)$

(13.22) There is an ant that bites.
$(\Sigma x \in \text{ant})\ \text{bites}(x)$

Given a pair $\langle a, b\rangle$ from $(\Sigma x \in A)B$, we can apply *left* and *right projections* l and r to extract the left component

$$l\langle a, b\rangle = a$$

and the right component

$$r\langle a, b\rangle = b.$$

Noting that we can encode two variables in one (via pairing $\langle \cdot, \cdot \rangle$), let us return to Geach's donkey sentence (13.17), expressing implication through Π.[4]

(13.17) If (a farmer)^x owns (a donkey)^y he$_x$ beats it$_y$.

The antecedent, (*a farmer*)^x *owns* (*a donkey*)^y, is translated to the type

$$(\Sigma x \in \text{farmer})(\Sigma y \in \text{donkey})\text{owns}(x, y)$$

using Σ to existentially bind the variables in the type expression owns(x, y). We connect the antecedent to the consequent through Π, turning (13.17) into the type

$$(\Pi z \in (\Sigma x \in \text{farmer})(\Sigma y \in \text{donkey})\ \text{owns}(x, y))\text{beats}(lz, l(rz))$$

where the subscripts x and y in (13.17) become terms lz and $l(rz)$ built from the left and right projections l and r. Navigating with l and r undeniably obscures the variable co-indexing in (13.17).

We can improve transparency by reformulating Σ in terms of records, as advocated in Cooper (2005), so that the type for (13.17) becomes

$$(\Pi z \in \begin{bmatrix} x : \text{farmer} \\ y : \text{donkey} \\ s : \text{owns}(x, y) \end{bmatrix})\text{beats}(z.x, z.y)$$

with : in place of $\in$, and with dotted projections $z.x$ and $z.y$ in place of lz and $l(rz)$. Within records, x and y above are called *labels*, rather than variables. But the variable/label distinction is a purely technical one that we can ignore as soon as we understand it, in much the same way that we can confuse variables with constants so as to construe the pair (M, f) of a model M and variable assignment f as a model expanding M. Treating labels as variables, a record type

$$\begin{bmatrix} x_1 : T_1 \\ x_2 : T_2(x_1) \\ \vdots \\ x_n : T_n(x_1, x_2, \ldots, x_{n-1}) \end{bmatrix}$$

amounts to a finite sequence (13.23) of variables paired with types, as mentioned in (ii) above.

(13.23) $x_1 : T_1,\ x_2 : T_2(x_1),\ \ldots\ x_n : T_n(x_1, x_2, \ldots, x_{n-1})$

Martin-Löf type theory builds particular sequences of the form (13.23), called *contexts*. To avoid confusion with other notions of context, let us refer to these as *t-contexts*. We take the empty sequence as a t-context (assigning no variable a type), and generate t-contexts inductively according to

[4] Similarly, conjunction can be expressed through Σ (inasmuch as a proof of $A \wedge B$ is a pair consisting of a proof of A and a proof of B).

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 295 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

(∗) given a t-context Γ and a variable x not already assigned a type by Γ, we can form the t-context

$\Gamma,\ x : T$

(assigning x the type T) provided that T is a type relative to Γ.

Let us agree to write

$\Gamma \vdash T$ type

to express the proviso in (∗) that T is a type relative to the t-context Γ. We have, for Q equal to Σ or Π, the rule

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash (Qx \in A)B \text{ type}}\ x \text{ not in } \Gamma$$

stating that if

A is a type relative to Γ and B is a type relative to $\Gamma, x : A$

then $(Qx \in A)B$ is a type relative to Γ. Consider again (13.16).

(13.16) a. (Pat's car)$_x$ is yellow.
b. Pat has (a car)^x. It$_x$ is yellow.
c. Pat has (a car)^x. (Pat's car)$_x$ is yellow.
d. If Pat has (a car)^x, it$_x$/(Pat's car)$_x$ is yellow.

Assuming

$x_1 : (\Sigma x \in \text{car})\ \text{pat-has}(x) \vdash \text{yellow}(lx_1)$ type

we can derive

$\vdash (\Pi x_1 \in (\Sigma x \in \text{car})\ \text{pat-has}(x))\ \text{yellow}(lx_1)$ type

for (13.16d), or

$\vdash (\Sigma x_1 \in (\Sigma x \in \text{car})\ \text{pat-has}(x))\ \text{yellow}(lx_1)$ type

for a conjunctive form

Pat has (a car)^x, and it$_x$ is yellow

of (13.16b), or

$x_1 : (\Sigma x \in \text{car})\ \text{pat-has}(x), x_2 : \text{yellow}(lx_1)$

for (13.16bc).[5]

[5] For simplicity, we ignore here the difference between pronouns and definite descriptions. The interested reader is referred to Fernando (2001) for a more detailed proof-theoretic treatment of presupposition, linked to conservativity of generalized quantifiers.

It follows from (∗) that t-contexts of the form (13.23) satisfy

(nov) $x_i \neq x_j$ for $i \neq j$

and

(fam) if x_i occurs in T_j then $i < j$

for i and j ranging over $\{1, 2, \ldots, n\}$. The consequences (nov) and (fam) correspond respectively to novelty and familiarity requirements on discourse referents, and together support a DRT-like treatment of anaphora. But whereas a string $K_1 \cdots K_n$ of DRSs K_i representing an n-sentence text $S_1 \ldots S_n$ is flattened by merge $\uplus$ to a single DRS

$K_1 \uplus \cdots \uplus K_n = (U, C)$

consisting of a set U of discourse referents and a set C of conditions, the sequential structure $S_1 \ldots S_n$ is preserved if we represent each sentence S_i by a type $T_i(x_1, \ldots, x_{i-1})$ relative to a t-context

$x_1 : T_1,\ \ldots,\ x_{i-1} : T_{i-1}(x_1, \ldots, x_{i-2})$.

Under this formulation, discourse referents arise as terms assembled from the t-context relative to which a type is defined. The t-context can include assumptions such as

all cars have engines
$(\Pi y \in \text{car})(\Sigma z \in \text{engine})\ \text{has}(y, z)$

allowing us to form a term for the engine mentioned in (13.19).

(13.19) Pat has (a car)^x. (The engine)$_y$ needs repair.
$x_0 : (\Pi y \in \text{car})(\Sigma z \in \text{engine})\ \text{has}(y, z)$,
$x_1 : (\Sigma x \in \text{car})\ \text{pat-has}(x)$,
$x_2 : \text{needs-repair}(l(x_0(lx_1)))$

Terms formed from such proofs (based on additional assumptions) may far outstrip the discourse referents enumerated in a DRS, and must be formed with restraint if we are to avoid overgeneration.

13.3.2 Denotations and instances

To instantiate the variables in a t-context, we interpret type expressions alongside t-contexts. More precisely, for each t-context Γ and type expression T such that $\Gamma \vdash T$ type, we define the notion of an *instance s of* Γ and the *denotation* T_s *of* T *relative to s* as follows.

(a) For Γ equal to the empty sequence, there is exactly one instance of Γ, the empty map $\emptyset$.

(b) T_s is the type obtained from the type expression T by replacing every variable x_i in T by $s(x_i)$.
(c) If Γ does *not* assign a variable x a type, then an instance of $(\Gamma, x : T)$ is an instance s of Γ expanded to map x to some object a in T_s.

Implicit in (b) and (c) is the assumption that T_s is a well-defined set whenever $\Gamma \vdash T$ type and s is an instance of Γ.[6] For T's built with neither Σ nor Π, various choices can be made (just as a DRS can be interpreted against any number of models). Fixing some such choice, let $\mathsf{char}(T)$ be the function that maps an instance s of a t-context Γ such that $\Gamma \vdash T$ type to the denotation

$$\mathsf{char}(T)(s) = T_s$$

of T relative to s. The similarity between $\mathsf{char}(T)$ and $\mathsf{character}_{\mathcal{M}}(\varphi)$ from Section 13.2.1 is unmistakable (see Table 13.2). Two important differences, however, should be noted. First, whereas $\mathsf{content}_{\mathcal{M}}(\varphi, f)$ in Section 13.2.1 does not incorporate additions to f (arising from discourse referents φ takes to be novel), the denotation T_s encodes these additions directly into its objects (proofs).[7] Secondly, the notion of a t-context Γ structuring $\mathsf{char}(T)$ abstracts away the particular choices (M, f) in Section 13.2.1 and 13.2.2, providing a model-free semantic account. Familiar constructs in typed programming languages, typings of variables (in a t-context) are hypothetical until they are executed (as programmes are intended). Although t-contexts allow us to steer clear of models, a notion of reference is crucial for hooking up type expressions with an external reality. Situation semantics is linked, for instance in Cooper (2005), to Martin-Löf type theory on the basis of the following reading of Austin (1961) in Barwise and Perry (1983)

Table 13.2 Section 13.2.1 vs. Section 13.3.2

Section 13.2.1	Section 13.3.2
formula φ	type T
$\mathsf{character}_{\mathcal{M}}(\varphi)$	$\mathsf{char}(T)$
$\mathsf{content}_{\mathcal{M}}(\varphi, f)$	denotation T_s relative to s
variable assignment f	instance s of a t-context

[6] The notion of set here can be understood in the sense of ordinary set theory (e.g. ZFC) under classical logic, assuming we formulate the dependent types $(\Sigma x \in A)B$ and $(\Pi x \in A)B$ as in, e.g., Feferman (1975).

[7] In programming language terms applied to 'apparently noncompositional phenomena in natural languages' in Shan (2005), the introduction of a discourse referent is a *side-effect* of evaluating an existential formula under dynamic formulations of DRT (based on quantified dynamic logic). By contrast, the type-theoretic approach considered here incorporates witnesses into the (constructive) proofs of an existential formula (Σ-type), and does so within a declarative rather than an imperative (assignment-based) programming language.

> a statement is true when the actual situation to which it refers is of the type described by the statement. (Barwise and Perry, 1983: 160)

Austin distinguishes *demonstrative conventions* that determine reference from *descriptive conventions* that yield, in the present framework, type expressions. Without fixing either the demonstrative or descriptive conventions of a language, instances s of t-contexts together with denotations T_s of type expressions T give us a foothold by specifying which descriptions apply to which references. These specifications are compositional insofar as (PC) holds for a t-context as a PC-input, and its set of instances as a PC-output. That is, we may represent a sequence of n sentences not by a string $K_1 \cdots K_n$ of n DRSs (as in Table 13.1, Section 13.2.2) but by a t-context of the form (13.23).

(13.23) $x_1 : T_1,\ x_2 : T_2(x_1),\ \ldots\ x_n : T_n(x_1, x_2, \ldots, x_{n-1})$

Let us define the *interpretation* $[\![\Gamma]\!]$ *of* a t-context Γ to be its set of instances. By definition,

Proposition 2. *Whenever* $\Gamma \vdash T$ type *and* x *is a variable,*

$[\![\Gamma, x : T]\!] = \{s \cup \{(x, a)\} \mid s \in [\![\Gamma]\!]$ *and* $a \in T_s\}$

provided Γ *does not assign* x *a type.*

An immediate corollary of Proposition 2 is that the interpretation $[\![\Gamma]\!]$ of a t-context Γ is compositional. In accordance with assumption (b2) from Section 13.1.1, the PC-output s feeds into the interpretation of the PC-input $\Gamma, x : T$ (or more specifically, the type expression T). By linking meanings ever more tightly with contexts, we blur the line between semantics and pragmatics, bringing phenomena such as presupposition within the compass of compositional interpretation. One of the pay-offs in taking (13.23), rather than $K_1 \cdots K_n$, as a PC-input is the perspicuous representation of anaphoric connections by occurrences of x_i in T_j for $1 \leq i < j \leq n$.

13.4 Representation and structure

Discourse interpretation in the previous sections proceeds from a representation of a sequence $S_1 \ldots S_n$ of n sentences by a string of $K_1 \cdots K_n$ of n DRSs (in Section 13.2.2) or (in Section 13.3.1) by a t-context of the form (13.23) with length n. The obvious question is how well does either picture represent discourse? For example, some instances of (13.6a) are, as noted in the introduction above, often analysed as (13.6b) or as (13.6c).

(13.6) a. $S.S'$.
b. S because S'.
c. S and then S'.

We turn in the present section to conceptions of discourse structure beyond that suggested by the sequence $S_1 \ldots S_n$. As it happens, (13.23) can encode more structure than that exploited in the previous section. Indeed, (13.23) can serve as a representation of

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 299 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

texts of m sentences where m differs from the length n in (13.23). This has consequences for denotations of type expressions that we explore below.

13.4.1 PC-inputs constrained and reconstrued

One reason for preferring (13.6b) or (13.6c) to (13.6a) is the intuition that *not* every sequence of interpretable sentences constitutes a coherent discourse, but that at the very least, the sentences in the sequence must have something to do with each other. Just what that something is is spelled out in (13.6b) and (13.6c), and represented in (13.24) through a type expression in which a variable from the t-context occurs.

(13.24) a. S. That is because S'.
$x : T,\ x' : T'$-explaining(x)
b. S. After that, S'.
$x : T,\ x' : T'$-following(x)

The type expressions T and T' in (13.24) correspond to the sentences S and S' respectively (according to what Austin calls descriptive conventions). Without worrying for the moment about how to choose a type expression, let us focus on the occurrence of a variable in it. Given a t-context Γ of the form

$x_1 : T_1,\ \ldots,\ x_n : T_n$

(as in (13.23)), let us collect all pairs (i, j) of indices such that x_i occurs in T_j in

$\mathsf{Link}_\Gamma = \{(i, j) \mid 1 \leq i, j \leq n \text{ and } x_i \text{ occurs in } T_j\}$.

It follows from familiarity (one of the consequences of assuming Γ is a t-context)

(fam) if x_i occurs in T_j then $i < j$ (where $1 \leq i, j \leq n$)

that $i < j$ whenever $\mathsf{Link}_\Gamma(i, j)$. Next, let Conn_Γ be the symmetric, transitive closure of Link_Γ. Conn abbreviates *conn*ected, the idea being that $\mathsf{Conn}_\Gamma(i, j)$ holds precisely when the sentences S_i and S_j represented in Γ are connected by variables occurring in type expressions. The requirement that the sentences in a coherent discourse are connected corresponds (in the representation Γ) to the constraint

(con) whenever $1 \leq i < j \leq n$, $\mathsf{Conn}_\Gamma(i, j)$.

The t-contexts Γ in (13.24a) and (13.24b) respect (con). Notice that every prefix of a t-context is a t-context but that a prefix of a t-context respecting (con) need not respect (con). (For an illustration, recall lines (13.11) and (13.12), as discussed in the introduction.) Accordingly, we should be careful about imposing (con) on every PC-input, leaving (con) instead as a final check once the sequence is complete. Otherwise, an incremental interpretation of (13.23) that processes typings $x_i : T_i$ one at a time from $i = 1$ to $i = n$ may fail.

That said, there are ways of building t-contexts from (13.23) other than by appending a typing

$x_{n+1} \ : \ T_{n+1}$

to its right. For an illustration, consider the discourse (13.25), analysed in Asher and Lascarides (2003) using *Segmented DRT* (SDRT).

(13.25) a. John had a lovely evening.
b. He had a great meal.
c. He ate salmon.
d. He devoured lots of cheese.
e. He won a dancing competition.

SDRT extends DRSs to *Segmented DRSs* with rhetorical relations such as *Narration* and *Elaboration* linking *discourse segments*. Reformulating the SDRT analysis of (13.25) in terms of t-contexts and types, the five sentences are individually assigned type expressions T_a, T_b, T_c, T_d, and T_e that combine *not* into a flat 5-variable t-context of the form

$x_a : T_a, \ \ x_b : T'_b, \ \ x_c : T'_c, \ \ x_d : T'_d, \ \ x_e : T'_e$

(decorating T_b, T_c, T_d, and T_e with primes to allow for modifications by rhetorical relations) but instead into a 3-variable t-context

$x_a : T_a, \ \ y_b : U_b, \ \ z_b : \mathit{Elaboration}(x_a, y_b)$

where U_b is the record type

$$U_b = \begin{bmatrix} x_b : T_b \\ y_c : \begin{bmatrix} x_c : T_c \\ x_d : T_d\text{-}\mathit{Narration}(x_c) \end{bmatrix} \\ z_c : \mathit{Elaboration}(x_b, y_c) \\ x_e : T_e\text{-}\mathit{Narration}(x_b) \end{bmatrix}$$

representing the discourse segment (13.25b)–(13.25e). Discourse segments of more than one sentence arise from *subordinating* rhetorical relations such as *Elaboration*, as opposed to *coordinating* rhetorical relations such as *Narration*. The typing

$z_b : \mathit{Elaboration}(x_a, y_b)$

relates (13.25a) to the discourse segment (13.25b)–(13.25e) by *Elaboration*, just as in U_b, the typing

$z_c : \mathit{Elaboration}(x_b, y_c)$

relates (13.25b) to the discourse segment (13.25c)–(13.25d) by *Elaboration*. In U_b, *Narration* links (13.25c) to (13.25d), and (13.25b) to (13.25e). Note that a simple narrative sequence $S_1 \ldots S_n$ would result in a flat t-context

$x_1 : T_1, \ x_2 : T_2\text{-}\mathit{Narration}(x_1), \ \ldots \ x_n : T_n\text{-}\mathit{Narration}(x_{n-1})$

978–0–19–954107–2 14-Werning-c13 Scherer (Typeset by SPi) 301 of 306 November 17, 2011 21:47
OUP UNCORRECTED PROOF – REVISES, 17/11/2011, SPi

of the same length n. Subordinating rhetorical relations introduce a hierarchical structure (built above with variables/labels y_b, z_b, y_c, and z_c) that strongly violates the associativity in flat sequences.[8] The t-context for (13.25) grows incrementally from

$x_a : T_a$

after (13.25a), to

$x_a : T_a,\ y_b : [x_b : T_b],\ z_b : \mathit{Elaboration}(x_a, y_b)$

after (13.25b), to

$$x_a : T_a,\ y_b : \begin{bmatrix} x_b : T_b \\ y_c : [\, x_c : T_c \,] \\ z_c : \mathit{Elaboration}(x_b, y_c) \end{bmatrix}, z_b : \mathit{Elaboration}(x_a, y_b)$$

after (13.25c), to

$$x_a : T_a,\ y_b : \begin{bmatrix} x_b : T_b \\ y_c : \begin{bmatrix} x_c : T_c \\ x_d : T_d\text{-}\mathit{Narration}(x_c) \end{bmatrix} \\ z_c : \mathit{Elaboration}(x_b, y_c) \end{bmatrix}, z_b : \mathit{Elaboration}(x_a, y_b)$$

after (13.25d) and finally to

$x_a : T_a,\ y_b : U_b,\ z_b : \mathit{Elaboration}(x_a, y_b)$

after (13.25e). Without necessarily increasing the length of the (outer) t-context by one, each sentence adds to the overall structure according to a so-called *right frontier constraint*. SDRT appeals to this constraint, for instance, to explain the unacceptability of continuing (13.25a)–(13.25e) with (13.25f).

(13.25) f. ?It was a beautiful pink.

Because the salmon is not on the right frontier, it is not available for the pronoun *it* in (13.25f). A definite description can, however, access it (using, as van der Sandt (1992) observes, its greater descriptive content).

(13.25) f′. The salmon gave him all the energy he needed.

Exactly what discourse constraints hold when is a delicate matter subject to debate. Discourse adverbials such as *then*, *instead*, and *otherwise* are argued in Webber et al. (2003) to be anaphors exempt from constraints on 'structural' conjunctions such as *and*, *but*, and *although*. It is straightforward to represent discourse adverbials in the framework of t-contexts. The framework is neutral on the question of what constraints to impose or of precisely what rhetorical relations to use (e.g. Kehler, 2002), if any (as some semanticists are loathe to use them). Beyond restricting the domain of compositional interpretation in discourse,[9] such questions have consequences for the interpretation

[8] Notice that the constraint (con) above ensuring that the sentences in a discourse are connected must be modified to account for the hierarchical structure.

[9] Recall from the introduction that one way to account for example (13.1) without fiddling with PC-outputs is to restrict PC-inputs.

of subsentential units, crossing (and thereby challenging) the divide between multi-sentential and subsentential processing. It is a trivial matter to turn the analysis (13.24a) of two sentences into one for the single sentence (13.6b), *S because S′*,

$$y : \begin{bmatrix} x & : T \\ x' & : T'\text{-explaining}(x) \end{bmatrix}$$

(or perhaps

$$y : \begin{bmatrix} x & : T \\ y' & : \begin{bmatrix} x' & : T' \end{bmatrix} \\ z' & : \mathit{Explanation}(x, y') \end{bmatrix}$$

with a subordinating rhetorical relation *Explanation*). But it is far from obvious that t-contexts and the dependent type constructs Π and Σ (perhaps reformulated in terms of records) suffice for discourse, in its multi-sentential and subsentential splendour. Even so, it is worth noting how far they go in satisfying a minimal assumption about PC-inputs stated in Section 13.1.1.

(b1) PC-inputs have enough structure to associate truth-conditions with them.

Beyond representing anaphoric connections by occurring in type expressions, the variables in a t-context serve as useful labels (written $\pi, \alpha, \ldots$ in Asher and Lascarides (2003)) through which to express various abstract objects that figure in PC-outputs. We turn to some of these next.

13.4.2 Decomposing PC-outputs

What exactly does a variable x_i in a t-context (13.23) represent? Consider again (13.24a).

(13.24) a. *S*. That is because *S′*.
$x : T,\ x' : T'$-explaining(x)

It is natural to assume that part (if not all) of what x and x' represent are semantic entities e_x and $e_{x'}$ such that the relation of explanation between x and x' can be cashed out as a causal relation between e_x and $e_{x'}$. The semantic entities e_x and $e_{x'}$ are called situations *characterized by S* and *by S′* (respectively) in Schubert (2000). To account for causation, Schubert argues that not only must a sentence be true in a situation it characterizes, but that the sentence must be about the situation as a whole, with no part of the situation extraneous to the sentence. (For instance, a situation characterized by the sentence 'Pat walked to the train station' should not include any subsequent train rides that Pat took.) As a consequence, possible worlds are ruled out as characterized situations. That is, situations are used as finer grained alternatives to total (fully fleshed out) worlds (much in the spirit of, for example, situation semantics, Barwise and Perry (1983)).

The tight fit between a situation and a sentence that characterizes it is similar to that assumed between an event and a predicate on the event in event semantics (e.g. Davidson, 1967*a*; Parsons, 1990). Indeed, the situations e_x and $e_{x'}$ are called the *main*

eventualities of x and x' (respectively) in SDRT, with 'eventualities' understood to cover events, states, and the like. The semantic significance of a rhetorical relation typically comes down to some relation between the corresponding main eventualities. If the semantic content of z such that

$z : \mathit{Explanation}(x, y)$

is that the main eventuality e_y of y causes the main eventuality e_x of x, then for

$z : \mathit{Elaboration}(x, y)$

it is that e_y is part of e_x (Asher and Lascarides, 2003: 160). The notions of causation and part-of are not straightforward, and have potentially far-reaching ramifications for compositionality in discourse, especially if discourse structure is to stretch from the multi-sentential to the subsentential.

In Reichenbach's (1947) influential analysis of tense and aspect, the event time E is compared not only to the speech time S, but also to a reference time R, which is used, for instance, to distinguish the simple past from the present perfect (e.g. Steedman, 2000).

(13.26) a. Pat left Dublin but is back.
b. Pat has left Dublin ?but is back.

PC-outputs for discourse might similarly encode not only the main eventuality (described by Asher and Lascarides, (2003) as 'the semantic index of a clause' in head-driven phrase structure grammar (HPSG)) but also discourse analogues of S and of R pertaining to utterance and information structure, respectively. Information structure has to do with *how* content (or denotation, as in Table 13.2, Section 13.3.2 above) is packaged, negotiated, or managed through focus, topic, questions, and the like. It turns out that a sentence's truth-conditional content may depend on focus, which in turn is tailored to the question it is understood to answer (Rooth, 1985).

(13.27) a. Mary only introduced $[\text{Bill}]_F$ to Sue.
(a reply to: Who did Mary introduce to Sue?)
b. Mary only introduced Bill to $[\text{Sue}]_F$.
(a reply to: Who did Mary introduce Bill to?)

Focus is indicated in (13.27) by the subscript F, which marks an argument to a predicate that when replaced (i.e. abstracted out) by a variable, suitably restricted, induces the question answered (or 'background'), under a *structured meaning* account (e.g. Krifka, 2001*b*). The λ-abstraction involved here can be carried out in the type-theoretic framework of t-contexts, although the proper treatment of focus remains controversial (e.g. Beaver and Clark, 2003), as does that of questions (e.g. Ginzburg and Sag, 2000).

If information structure is analogous to Reichenbach's reference time R (just as the main eventuality is analogous to Reichenbach's event time E), what about speech time S? Ginzburg and Cooper argue that the semantics of dialogue requires utterances given by an

> Utterances as events hypothesis: Utterances are spatio-temporally located events involving the sequential enunciation of one or more words. (Ginzburg and Cooper, 2004: 298)

For data, they point to *clarification ellipsis* (CE), as in (13.28), which appears as (4) in Ginzburg and Cooper (2004).

(13.28) a. A: Did Bo finagle a raise?
B: (i) Bo?/(ii) Finagle?
b. **Clausal reading**: Are you asking if BO (of all people) finagled a raise/Bo FINAGLED a raise (of all actions).
c. **Constituent reading**: Who is Bo?/ What does it mean to finagle?

To analyse CEs, Ginzburg and Cooper propose a

> Hybrid content hypothesis: the content which is updated in dynamic semantics consists of structure expressing detailed relationships between the content and the formal properties (syntax, phonology, etc.) of the various parts of an utterance. (Ginzburg and Cooper, 2004: 298)

The hybrid content hypothesis, HCH, is a far cry from the assumption (a2) that confines meaning to truth conditions (or indeed from the modest steps away from (a2) taken in DRT). Meanings under HCH are far richer, but then so are the expressions that have these meanings. Ginzburg and Cooper (2004: 306) require utterance representations that are 'fractally heterogeneous' in that 'the requisite representation format needs to contain heterogeneous (viz. phonological, syntactic, semantic, and contextual) information and, moreover, this applies uniformly as the parts get smaller and smaller'.

Notice that in a CE, a previous utterance forms part of the reality that language is about. This is surprising only to the extent that we are accustomed to a clean separation between language and the world, despite the existence of words (such as 'aforementioned' and 'latter') and phrases (such as 'that said' or 'replacing the second syllable of the last mentioned word by the third syllable of the first') that treat language as part of the world language describes or queries. Lest we dismiss such metalinguistic constructions as marginal, Ginzburg and Cooper (2004: 299) point out that CEs 'are commonplace in human conversation'. That said, there is more to the world that language is about than language. And even when language enters into that world (and knowledge of language becomes part of world knowledge, which is used say, to interpret definite descriptions or rhetorical connections between sentences), it may well prove helpful to distinguish language as a means of communication from language as what communication is about. In any case, the temptation to push some form of Montogovianism as far as it will go continues to be irresistible for many formal semanticists. And the proper integration of world knowledge and linguistic knowledge remains a vexed question.[10]

[10] My thanks to my referees for their helpful criticisms and advice.