account of the logical and truth-conditional properties of sentences.

All forms of discourse-based semantics are based on the central assumption that comprehension involves the mental construction by a comprehending subject C of a representation of some, not necessarily real, state of affairs, to any degree of detail, on the basis of linguistic material (uttered sentences) produced by a producer P, who expresses his own representation. When the two representations coincide, comprehension has been adequate. P expresses his representation piecemeal, by an orderly sequence of uttered sentences, each of which contributes to C's building up of his representation. The process of successively building up discourse representations is called 'incrementation.' An adequate logical model-theory, or specification of truth-conditions, should, in this perspective, not take as its input sentence structures (of any level of syntactic depth) but, rather, well-structured discourse representations.

In DSX, a representation of this kind is called a 'discourse domain' (D). The incrementation of a sentence A, or i(A), to some D results in the storage in D of the semantic information carried by A. This is done by setting up representations ('addresses') for the individuals or sets of individuals mentioned in the discourse, and subsequently distributing the information made available by newly added sentences over the addresses concerned. Besides addresses, a D may contain 'instructions,' which constrain its further development. Negation of a sentence A, for example, results in an instruction to block the incrementation of A in D. In this perspective, the linguistic meaning of a sentence A is conceived of as the contribution of i(A) to any given D in virtue of A's systematic linguistic properties.

The ancillary status of sentence meaning with respect to the construction of superordinate Ds is reflected mainly by three kinds of phenomena, namely identification across intensional domains, anaphora phenomena, and presuppositions. All three have led to a separate approach to discourse-based semantics. Yet despite the differences, there is a common core of ideas. Identification phenomena prompted Fauconnier to develop his theory of mental spaces (1985). Anaphora phenomena (after a first shot in Isard (1975)) led to discourse representation theory (DRT) (see, for example, Kamp 1981; Heim 1990; Kamp and Reyle in press), and presuppositions to DSX, which is at issue in this article (Seuren 1972, 1975, 1985; Gazdar 1979).

Identification across intensional domains is illustrated by considering a sentence like *The girl with brown eyes has blue eyes.* This sentence seems inconsistent and hence uninterpretable, yet it makes perfect sense if interpreted, for example, as 'the girl represented in the picture with brown eyes in reality has blue eyes,' or as 'the girl who in reality has brown eyes is represented in the picture as having blue eyes.' In other words, the context may provide a means of assigning a sensible interpretation to this seemingly uninterpretable sentence. Fauconnier (1985) posits 'mental spaces,' autonomous but interconnected cognitive domains of interpretation whose elements are open to denotation by definite NPs under certain conditions. Anaphora phenomena being a subset of identification phenomena, this approach has elements in common with DRT. But Fauconnier also brings presuppositions into play. Although mental space theory

Discourse Semantics

'Discourse semantics' (DSX) is the name given to one of the varieties of discourse-based semantics that came into being during the 1980s. Like the other varieties, it is part of a realist theory of linguistic meaning, that is, a formal theory aimed at a reconstruction by hypothesis of the processes involved in and causing the comprehension of linguistic messages. It is therefore part of cognitive science, and distinguishes itself from the established discipline of formal semantics (see *Formal Semantics*), which is limited to an

has so far been presented only in relatively rough outline, it should provide a good basis for integration with DRT and DSX.

DRT is based on anaphora (see *Anaphora*). Anaphora occurs when an anaphoric element, usually a pronoun, takes over the (constant or variable) reference function of another expression, its 'antecedent,' whose reference function is independently grounded. In standard model-theory, an anaphoric pronoun occurring in a true extensional sentence must be analyzed as either a bound variable or an expression with (anaphorically related) constant reference. This gives rise to a problem, known as 'the problem of the donkey sentences' (after Geach 1962). It consists in the fact that some anaphoric pronouns cannot be analyzed as either bound variables or referring expressions. For example, even if a sentence like:

If Nob owns a donkey he feeds it. (1)

is true, the pronoun *it* does not have to refer to any given donkey, nor does it allow for regular binding by the existential quantifier expressed by *a donkey*. Note that binding by the universal quantifier, as in $\forall x[[\text{donkey}(x) \wedge \text{have}(\text{Nob}, x)] \rightarrow \text{feed}(\text{Nob}, x)]$, lacks generality, in view of sentences like:

If Nob wants to own a donkey he will have to feed it. (2a)

If Nob finally owns a donkey he will probably feed it. (2b)

where this analysis again runs foul of scope problems.

The problem that donkey sentences pose for standard model-theory led to the development of DRT, which posits 'discourse representation structures' (DRSs) as an intermediate station between sentences and any 'world' they are interpreted in. Anaphoric relations are now said to hold not with respect to any such 'world,' but with respect to elements in DRSs. DSX and, presumably, also mental space theory fully subscribe to this solution.

DRT both continues and innovates on the standard, strictly bivalent, logical tradition as developed during the twentieth century. It continues, in particular, an earlier innovation, mainly due to Davidson and Montague, which came about in the 1960s and consisted in viewing natural languages as formal languages, that is, with a precisely defined syntax and susceptible of logical model-theory. DRT follows Montague in taking the surface structure of sentences as input to the model-theoretic interpretation procedure. This is precisely what Montague intended when viewing English as a formal language: surface structures are taken to be directly susceptible of semantic interpretation, without the assumption of an intermediate level of semantic analysis expressing meanings regularly and unambiguously in some variety of modern predicate calculus. Instead of a semantic analysis formulated in terms of predicate calculus, DRT posits DRSs as an intermediate level in the process of semantic interpretation of (surface structures of) sentences onto possible worlds (for a full account of DRT see *Discourse Representation Theory*).

Besides anaphora, the subordination of sentence meaning to the setting up of complex Ds is reflected in the fact that most sentences carry one or more presuppositions (see *Presupposition*). A presupposition A of a sentence B is a systematic semantic property of B restricting its usability to those contexts to which A has already been incremented or admitting the incrementation of A without either logical or inductive (that is, knowledge-based) incompatibility. Other than DRT, which has not provided a treatment of presupposition so far—though Heim (1982) suggests (incorrectly as will be shown in Sect. 2.2) that presupposition should be taken as a specific form of anaphora—DSX takes presuppositional phenomena as its starting point. Among the first to stress the discourse function of presuppositions were Seuren (1972, 1975) and Gazdar (1979).

As for anaphora, the view is that anaphoric and other referential expressions refer not directly (this it has in common with DRT) but via addresses, i.e., 'discourse representations' of the (sets of) individuals referred to. The treatment proposed for implications and disjunctions is such that they may involve addresses that need not correspond to real world entities for truth. What counts is not reference but denotation, i.e., the relation between definite NPs and their addresses in D.

DSX continues an alternative tradition of logico-semantic analysis, whose origins lie in certain aspects of Frege's work and which was taken up by Strawson and others after 1950. This tradition drops strict bivalence as an axiom of the logical analysis of sentences, replacing it by either gapped bivalence or trivalence. It also rejects Russell's (1905) analysis of definite descriptions in terms of existential quantification, leaving them intact as legitimate argument terms of predicates.

A prominent feature of DSX is the rejection of 'surface semantics.' Other than DRT and the Montague tradition, DSX regards surface structures of sentences as in principle unfit for direct semantic interpretation. What is needed is a grammar relating each surface structure to its semantic analysis (SA), which is formulated in terms of a variety of modern predicate calculus (with generalized quantifiers). The SA of each sentence is taken to be the input to the incrementation procedure. The motivation behind the rejection of surface semantics lies in the fact that the surface form of sentences all too often conceals their true meaning. For example, many languages show 'spreading' (or 'copying') of semantically relevant elements. Negation copying, as in the Cockney sentence:

'E ain't never been no good to no woman, not never. (3)

is rife in the languages of the world. Clearly, the true meaning of this sentence does not correspond to its 'literal' logical analysis, i.e., 'he has sometimes been good to some woman.' What has happened is that the one, important, negation is copied a number of times without any semantic effect. The grammar of Cockney English specifies under what conditions such copying must or may take place. It follows that this copying must be 'undone' before the sentence is semantically interpreted. Given the multitude of cases where surface forms appear to conceal or distort their meanings (see Seuren 1985: 61–110), it seems good strategy to insert a grammar (G) between surface sentences and any machinery of semantic interpretation.

DSX thus posits, in effect, two intermediate stages between surface structures (SS) of sentences and whatever they may be taken to refer to in any possible world, i.e., the linguistically structured SA and the cognitively structured discourse domain (D), which is directly linked to a knowledge base (KB) of background (encyclopedic) and situational

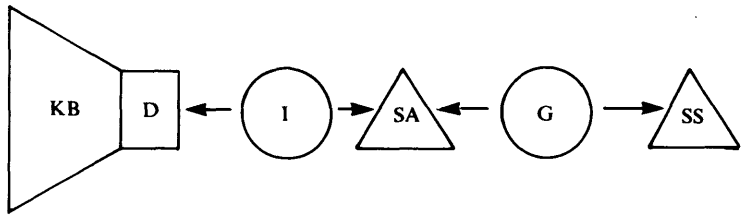


Figure 1.

knowledge. A set of incrementation rules (I) relates SAs to Ds and vice-versa. The overall structure of the theory is presented in Fig. 1.

DSX is divided into two main sections, the 'theory of clausal incrementation' and the 'theory of domains and subdomains.' The former deals with the incrementation procedure of linguistic clauses to the appropriate (sub)domains (including quantification). The latter deals with the notion of proper sequentiality in a (sub)domain and the embedding of subdomains representing what is said to be the case under an intensional operator or what are said to be alternative possibilities (as in disjunctions or implications).

1. The Theory of Clausal Incrementation

1.1 Address Architecture and Model-Theory

A D consists of 'addresses,' and it may also contain 'instructions' constraining its later development. Addresses are of different types. The discussion in this article will be limited to addresses representing individuals or sets of individuals in any possible 'verification domain' (partial world) V (disregarding addresses representing matter, properties, facts, and all kinds of Meinongian objects). Individual addresses are marked by the initial capital 'E.' They are singular or plural, and either open or closed. An open singular (1st order) address is of the form $Ea_n|Pred(a_n)$, as in:

$$Ea_1|cat(a_1) \quad (4)$$

which is the incremental result of *There is/was a cat* (tenses are disregarded throughout). The element preceding the vertical stroke, in this case 'Ea_1,' is called the address head. Address heads can be used, in other addresses, as a label for the address they head. Lower case variables (a_n) range over individuals, capitals (A_n) over sets of individuals.

D-structures are the proper formal objects for model-theoretic intepretation. In (4), '$cat(a_1)$' is a propositional (characteristic) function from individuals to truth-values, that is, of the type (e, t), referring to the set of all cats in the verification domain V. 'Ea_1' is a function from sets to truth-values, that is, typed ((e, t), t), delivering truth just in case the input set is nonempty. So far, the model-theoretic interpretation of (4) thus runs exactly parallel to that of the Russellian formula '$\exists x[cat(x)]$'. The propositional function '$cat(a_1)$' in (4) is called a 'predication,' the term used for any propositional function or full proposition stored in an address.

An open plural (2nd order) address is normally of the form $EA_n|::Pred(a_n)$, as in:

$$EA_1|::cat(a_1) \quad (5)$$

representing *There are/were cats.* The expression '$::cat(a_1)$' reads intuitively as 'for each individual *i* of the set whose existence is asserted by (5), *i* is a cat.' Formally, the operator '::' is a function from any set of elements A to the 'plural power set' of A or $P_{pl}(A)$, i.e., P(A) minus $\varnothing$ and all singleton sets. (At least as a default value; it can be overridden to the 'singular power set of A or $P_{sg}(A) = P(A) - \varnothing$, as when the answer to the question *Does Carol have children?* is *Yes* and Carol has only one child. Only plural power sets will be considered here.) As '$cat(a_1)$' is typed (e, t), '$::cat(a_1)$' is typed ((e, t), t), which makes (5) a 2nd order address. Open plural address heads are type-raised to (((e, t), t), t), that is, functions from sets of sets to truth-values, but with the same satisfaction condition: the input set must be non-empty. The (default) truth-condition of (5) is thus that there be at least one set of cats with cardinality ≥ 2.

Open plural addresses, however, may also be of the form $EA_n|Pred(A_n)$, in analogy to singular addresses. This depends on the typing of *Pred.* If *Pred* is 1st order, typed (e, t), as in (5), the distributive operator '::' is needed. But if *Pred* is 2nd order, typed ((e, t), t), then the incremental result is $Pred(A_n)$, as in the 2nd order address:

$$EA_1|team(A_1) \quad (6)$$

representing *There is/was a team.* Clearly, the two forms can be combined, as in:

$$EA_1|::cat(a_1), disperse(A_1) \quad (7a)$$

$$EA_1|::cat(a_1), 3(A_1) \quad (7b)$$

representing, respectively, *Some cats dispersed* and *There were three cats.*

Mathematically, an address can be of any order, yet natural language imposes a limit. A 3rd order address like:

$$E\bar{A}_1|::team(A_1) \quad (8)$$

reads *There are/were teams.* '$::team(A_1)$' is typed (((e, t), t), t), which makes (8) a 3rd order address. Natural language has only few nouns with a 3rd order lexical meaning, standing for sets of sets of individuals. *League*, as in football competitions, looks like an example, as it stands for a set of teams. *There was a league* is thus represented as:

$$E\bar{A}_1|league(\bar{A}_1) \quad (9)$$

and *There were leagues* results in the 4th order address:

$$E\bar{A}_1|::league(\bar{A}_1) \quad (10)$$

No addresses of a higher than 4th order seem required for natural language.

An open address is 'closed' as soon as it is denoted in a subsequent clause by a definite (anaphoric) term. The effect

of address closure is that the address head is type-changed from a function to truth-values to a function to an entity or set of entities, their 'referents.' The referent of a closed address under the head Ea_n (or EA_n) is named Ra_n (or RA_n). Closure is symbolized by a double vertical line. Predications added after closure are no longer propositional functions, but full propositions with Ra_n (or RA_n) replacing the bound variable. If (4) is followed by a sentence like *The cat ran away*, the incremental result is given as:

$$Ea_1 | \text{cat}(a_1) \| \text{run away}(Ra_1) \quad (11)$$

'Ea_1' is now retyped from ((e, t), t) to ((e, t), e), whereby the predication before closure normally determines the input set typed (e, t). The predication 'run away(Ra_1)' is the proposition that the individual selected by Ea_1, that is the cat, is an element of the set of those who ran away.

If V contains one single cat, 'Ea_1' selects that as the cat referred to. With more cats the function ((e, t), e) will fail to yield a value. But natural language allows for the reference function to locate its object even if more than one entity satisfies the address before closure. In such cases the predications added after closure are 'called in' to help identify a unique reference object. Thus, if V contains two cats, one that ran away and one that did not, Ea_1 as in (11) will lead to the cat that ran away. Then, the proposition 'run away(Ra_1)' cannot but be true, since it serves to locate the reference object. Given such a V, the definite term *the cat* in a text *There was a cat. The cat ran away* is said to have 'nonspecific reference' (see Seuren 1985: 459–64). Nonspecific reference is, however, strongly marked and not preferred. Listeners will expect, normally speaking, that if a cat-address is introduced, as in (4) above, then closure of the address will allow reference fixing on precisely one cat in V. The inference from *There was a cat* to *There was precisely one cat* is thus licensed by the default assumption that open addresses can be closed in a nonmarked way. This makes a Gricean explanation in terms of conversational implicatures (based on the cooperation principle) superfluous.

When an open plural address $EA_n | :: \text{Pred}_1(a_n)$ is closed, 'EA_n' becomes a function from sets of sets to sets: (((e, t), t),(e, t)). The set selected, referred to by RA_n, is normally (that is, with specific reference) uniquely identifiable in V. The default case is that the function selects the largest of the set of sets. Normally, that is, *The children left* implies that all the children left. But external factors may interfere, such as when the largest set is held responsible in some sense, even though only a few of its individual members satisfy the predicate. Thus, *The mice have been at the cheese* need not imply that all the mice have, not even all the mice in the house. Likewise, for example, *The Americans were the first to land on the moon*, which does not imply that all Americans were the first to do so.

Note, moreover, that under these assumptions, closure of an address like (7b) above requires that there be one unique set of three cats in V, since if there are more, the plural power set of the set of cats in V will contain more than one set of three cats and the reference function will fail to yield a value. The apparatus thus provides a non-Gricean explanation for the default interpretation of *There were three cats* as 'there were exactly three cats.'

Plurals are difficult, and only some of the problems can be dealt with here. One such problem is the well-known distinction between distributive and group readings. These may differ truth-conditionally. For example, the sentence *The British voted for Major* may be true on the group reading, but false distributively. The sentence *The students taught themselves* is true distributively only if each of the students studied without a teacher, but for the group reading it is sufficient that one group of students taught another.

Group readings are rendered analogously to singular increments. Thus, given an open plural cat-address, as in (5) above, the addition *They ran away* (group reading) is incremented as in (12), analogously to (11):

$$EA_1 | :: \text{cat}(a_1) \| \text{run away}(RA_1) \quad (12)$$

To calculate the truth-value of 'run-away(RA_1)' it is necessary to type-raise the predicate *run away* from (e, t) to ((e, t), t), so that it can process an input of type (e, t). This means that the extension of *run away* in V contains not only individuals but also sets of individuals: the model is no longer a first-order structure. Such type raising is normally possible for 1st-order predicates that are not, in virtue of their meaning, restricted to individuals (such as *dog*, *tree*, *have a headache*, *intelligent*, etc.).

Distributive readings must be rendered differently. *They ran away* (distributively) is taken to be incremented as in (13). Given '$EA_1 | :: \text{cat}(a_1)$,' closure and i(they ran away) yields:

$$EA_1 | :: \text{cat}(a_1) \| [:: \text{run away}(x)](RA_1) \quad (13)$$

or 'the set of cats referred to by 'EA_1' is one of the sets of more than one individuals running away.' (The variables x, y, z, are used as 'handymen' for odd jobs not directly associated with special addresses. Like the variables $a_1, a_2, \ldots, a_n$ they range over V.) This enables us to express the difference between the two readings of, for example, *The children carried a bag*, which involves either one bag (carried by all children) or as many bags as there were children. The latter, distributive, reading is expressed in (14a), the former in (14b) (distributively) and in (14c) (group) (note that subaddresses are introduced within a given address; see below):

$$EA_1 | :: \text{child}(a_1) \| [:: Ea_i | \text{bag}(a_i), \text{carry}(x, a_i)](RA_1) \quad (i \text{ ranges over } RA_1) \quad (14a)$$

$$EA_i | :: \text{child}(a_1) \| Ea_7 | \text{bag}(a_7), [:: \text{carry}(x, a_7)](RA_1) \quad (14b)$$

$$EA_1 | :: \text{child}(a_1) \| Ea_7 | \text{bag}(a_7), \text{carry}(RA_1, a_7) \quad (14c)$$

It now becomes clear how important the closure operation is for simple empirical reasons. Its crucial function appears, for example, from the following pair of sentences:

Nob sent few letters that were rude. (15a)

Nob sent few letters, and they were rude. (15b)

These two sentences have clearly distinct truth-conditions (unaccounted for in most semantic theories). The difference is brought out by the closure operation, as is shown in (15c) and (15d), respectively, where 'Ra_1' refers to Nob:

$$EA_2 | :: \text{letter}(a_2), :: \text{rude}(a_2), :: \text{send}(Ra_1, a_2), \text{few}(A_2) \quad (15c)$$

$$EA_2 | :: \text{letter}(a_2), :: \text{send}(Ra_1, a_2), \text{few}(A_2) \| [:: \text{rude}(x)](RA_2) \quad (15d)$$

The next step is universal quantification, whose notorious complexity is best demonstrated with the help of some

examples. First, any semantic theory must account for the differences between plural *the* and *all* (*every*, *each*). *The Americans were first on the moon* is different from *All Americans were first on the moon*, and *The Belgians won the match differs* from *All Belgians won the match.* Compare also *The protesters were numerous* and **All protesters were numerous.* It thus seems that at least some group readings disallow universal quantification. But other (most) group readings are happy with *all* (though not with *every*): *All children dispersed* (*congregated*, *sat in a circle*, etc.). Note that for such cases the Russellian analysis $\forall x[Qx \rightarrow Mx]$ falls foul of type problems, as does the standard analysis in generalized quantification theory, where truth follows when the Q-set is contained in the M-set.

An apparatus that aims at accounting for such facts in a maximally regular way shall now be considered. Take a few simple examples. *All* (*the*) *mice escaped* is incremented as (16a). *All* (*the*) *mice dispersed* corresponds to (16b), and *The cat caught all* (*the*) *mice* to ($16c_1$) (D contains '$Ea_2|cat(a_2)$'). They read as 'the discourse-defined nonempty set of mice $M \in P_{pl}(\{x|escape(x)\})$,' '$M \in \{X|disperse(X)\}$,' and '$M \in P_{pl}(\{x|catch(the\ cat, x)\})$,' respectively. ($16c_2$) reads: 'the $cat \in \{x|catch(x, all\ the\ mice)\}$.' The notation '[Pred(x)]' is used to refer to the set of whatever individuals satisfy Pred or '$\{x|Pred(x)\}$.' Likewise, '[Pred(X)]' refers to the set of whatever sets satisfy Pred or '$\{X|Pred(X)\}$,' and '[: : Pred(x)]' for '$P_{pl}(\{x|Pred(x)\})$.'

$EA_1|::mouse(a_1) \| all([::escape(x)], RA_1)$ (16a)

$EA_1|::mouse(a_1) \| all([disperse(X)], RA_1)$ (16b)

$EA_1|::mouse(a_1) \| all([::catch(Ra_2, x)], RA_1)$ ($16c_1$)

$Ea_2|cat(a_2) \| all([::catch(Ra_2, x)], RA_1)$ ($16c_2$)

The quantifier *all* is thus treated as a higher order predicate that takes terms referring to sets as object, and terms referring to sets of sets as subject. It is satisfied (delivers truth) just in case the object set is a member of the subject set. This applies to all cases and all readings, and thus provides a unified solution to the type problem set by the Russell analysis or by standard generalized quantification theory. (This solution is in conformity with Montague's advice to generalize from the most difficult case.)

The semantic difference between *all* and plural *the* seems to be that *the* selects the set K of elements defined by the address head after closure, while the sentence says that K is a member of the set of sets L defined by the predicate. *All*, on the other hand, while also saying that $K \in L$, requires specifically that no member of K be left out with regard to the predicate symbolizing L. This makes *all* unsuitable for higher order predicates like *numerous*, but all right for other higher order predicates like *disperse* or *sit in a circle*. *Every*, as will be shown below, distinguishes itself from *all* mainly in that *every* does not allow for group readings. The differences between *every* and *each* are not touched upon here.

A predication may contain a reference taken from another address, as in *The cat caught the mouse*, with D containing $Ea_1|cat(a_1)$ and $Ea_2|mouse(a_2)$:

$Ea_1|cat(a_1) \| catch\ (Ra_1, Ra_2)$ (17)
$Ea_2|mouse(a_2) \| catch\ (Ra_1, Ra_2)$

Both addresses are closed, as both NPs are definite, and both receive the information provided by *The cat caught the mouse.* This means a reduplication of information storage, following from the principle that D distributes the information provided by the text over the addresses representing the (sets of) individuals at issue.

A sentence like *The cat caught a mouse* is incremented as follows, with D containing '$Ea_1|cat(a_1)$':

$Ea_1|cat(a_1) \| Ea_2|mouse(a_2), catch\ (Ra_1, a_2)$ (18)
$Ea_2|mouse(a_2), catch\ (Ra_1, a_2)$

Ea_1 now contains the open address Ea_2, saying that the cat (Ra_1) caught a mouse. An open address Ea_n, being a function to truth-values, can be stored under another address Ea_m as a predication, provided a_m or Ra_m occurs in at least one of the predications of Ea_n. A *cat caught a mouse* is represented as:

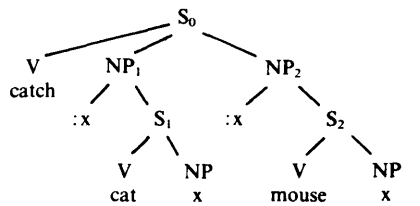
$Ea_1|cat(a_1), Ea_2|mouse(a_2), catch\ (a_1, a_2)$ (19)

The discourse may now continue with *the cat that caught a mouse*, closing Ea_1. Since it may also continue with a definite NP *the mouse*, an open address is needed of the form '$Ea_2|mouse(a_2)$, $Ea_1|cat(a_1)$, $catch\ (a_1, a_2)$.' This must be supplied by an auxiliary (logical) mechanism (not gone into here) of 'inferential bridging,' as is argued in the following section.

1.2 The Incrementation Procedure

The incrementation procedure I works top-down through the SA-input of any sentence to be incremented. For each S-structure, I starts with the highest predicate. Unless the predicate instructs otherwise, I increments the terms in left-to-right order. The most straightforward cases are those where a predicate is applied to definite terms, as in (20a). The grammar G assigns to (20a) the partial (i.e., without tense) SA (20b):

The cat caught the mouse. (20a)



(20b)

NP_1 reads as 'the x such that x is a cat'; analogously for NP_2. The incrementation of S_0, that is, $i(S_0)$, scans the predicate ('V') of S_0 first. *Catch* being a binary lexical verb, the standard procedure is followed: I is put to work on NP_1 first, to be followed by NP_2 ($i(NP_1)$ precedes $i(NP_2)$). D is deemed to contain '$Ea_1|cat(a_1)$' and '$Ea_2|mouse(a_2)$.'

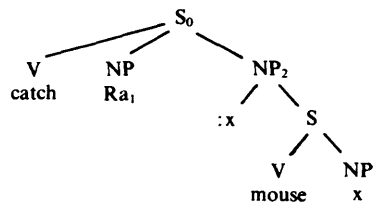
For any NP_i formed with the :-operator, $i(NP_i)$ consists of the following stages:

(a) The definite NP-operator ':' selects $d(NP_i)$, that is, the address Ea_n denoted by NP_i, taking as input the S-structure in its scope and selecting the matching address, which must, in principle, be unique in D.
(b) NP_i is replaced by 'Ra_n.'
(c) Ea_n is closed (if still open), and the entire SA-tree is added to the closed address, with 'Ra_n' for NP_i.
(d) Any definite NP_j occurring in an address predication is subjected only to stages (a) and (b): if $d(NP_j) = Ea_m$, NP_j is replaced by Ra_m in the predication S_j.

(For practical reasons trees are written as bracketed strings in predications.)

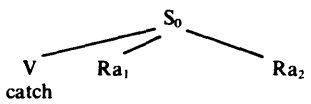
Thus, i(NP_1) in (20b) selects $Ea_1|cat(a_1)$, and NP_1 is replaced by Ra_1 in the SA-tree. Ea_1 is closed and the SA-tree, with Ra_1 in place, is added to the closed Ea_1:

$Ea_1|cat(a_1)\|$ (21)



NP_2 denotes Ea_2 and is replaced by Ra_2:

$Ea_1|cat(a_1)\|\|$



or, as a bracketed structure, '$Ea_1|cat(a_1)\|catch(Ra_1, Ra_2)$.' Analogously i($NP_2$) gives '$Ea_2 | mouse(a_2) \| catch(Ra_1, Ra_2)$.' The total result is thus as in (17) above.

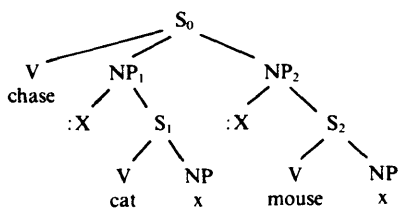
A comment on the denotation function is in order here. If D contains more than one address matching a definite NP, interpretation halts. If, on the other hand, the operator ':' fails to spot any matching address, the first step is to search background knowledge (with its defaults) to see if one of the existing addresses can still act as d(NP), as in:

Last night a Swiss banker was arrested at Heathrow Airport. The 53-year old bachelor declared that he had come to Britain to kidnap the Queen. (22)

Here, the NP *the 53-year old bachelor* clearly denotes the address just set up for the Swiss banker, a denotation that would have been impossible for, for example, *the officer in charge*. No formal procedure for this kind of identification is available as yet. If such identification is impossible, then, provided nothing in D or in background knowledge stands in the way, a new address is set up by 'suppletion post hoc' (or 'accommodation'; see *Accommodation and Presupposition*). If background knowledge does stand in the way of post hoc suppletion, interpretation halts.

When a plural NP is involved, there are usually two ways of incrementation, one for the group reading and one for the distributive reading. The difference must also be expressed in the input SA structure, if only to provide a syntactic analysis for readings like (14a) above. Assuming '$EA_1|::cat(a_1)$' and '$EA_2|::mouse(a_2)$' given in D, consider sentence (23a) with its SA (23b) representing the group reading ('the group of cats chased the group of mice'):

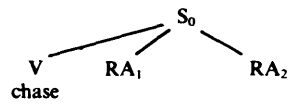
The cats chased the mice. (23a)



(23b)

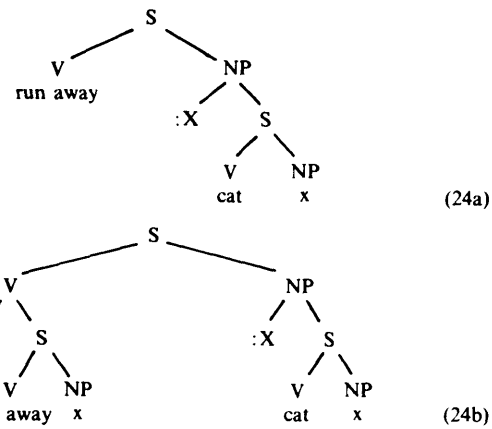
(NP_1 is to be read as 'the set X such that for each $x \in X$, x is a cat'; analogously for NP_2.) The incrementation proceeds analogously to the singular NPs: i(NP_1) selects '$EA_1|::cat(a_1)$' and is replaced by 'RA_1'; EA_1 is closed, and S_0 is added to EA_1, NP_2 being replaced by 'RA_2':

$EA_1|::cat(a_1)\|$



or: $EA_1 |:: cat(a_1) \| chase(RA_1, RA_2)$. Similarly: $EA_2 | :: mouse(a_2) \| chase(RA_1, RA_2)$.

The treatment of distributive plural readings is more complex. Let us revert to (13) above (i.e., '$EA_1 | :: cat(a_1) \| [:: run away(x)](RA_1)$'). The input SA of the increment '$[:: run away(x)](RA_1)$' is based on a variety of λ-extraction from the group reading SA (24a), given in (24b) (λ is replaced by the cap operator):



(24a)

(24b)

Now the main subject NP will select EA_1, close it and add (24b) with RA_1 for the subject NP, and with the ::-operator for ^X binding a lower case variable x.

If this procedure is applied to (23), the result is:

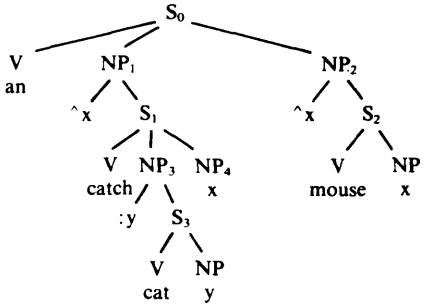
$EA_1|::cat(a_1)\|[_x::[_y::chase(x, y)](RA_2)](RA_1)$ (25a)

$EA_2|::mouse(a_2)\|[_y::[_x::chase(x, y)](RA_1)](RA_2)$ (25b)

The square brackets are subscripted to indicate the order in which the variables are to be rotated (i.e., substituted by names for all individuals in V): in $[_x::[_y::catch(x, y)](RA_2)]$ of (25a) the x-variable is rotated first, followed by the y-variable. For example, let there be three cats, three dogs, three mice, and three birds. First assign cat_1 to x and establish for each of the twelve individuals y whether cat_1 chased y. This gives $\{y|cat_1$ chased $y\}$. Now check if the set referred to by RA_2, that is, the three mice, is a member of $P_{pl}\{y|cat_1$ chased $y\}$. If so, assign the value '1' to cat_1. If not, assign '0.' Likewise for cat_2, cat_3, dog_1, dog_2, etc. for all twelve individuals. This gives the set of individuals x in V such that x chased the mice, e.g., $\{cat_1, cat_2, cat_3, dog_2, dog_3, bird_1\}$. Now check if the set referred to by RA_1, that is, the cats, is a member of $P_{pl}\{cat_1, cat_2, cat_3, dog_2, dog_3, bird_1\}$. If so, the sentence (increment) is true ('1'); if not, false ('0'). The reader notices that in the example the sentence is true. Analogously for (25b), which is, of course, also true. Note that the truth-checking for the distributive reading is based on a 1st-order model structure, whereas the group reading requires a higher order model structure (with sets as reference values for 1st-order predicate terms).

Existentially quantified sentences are treated as follows. Take (26a) with SA (26b):

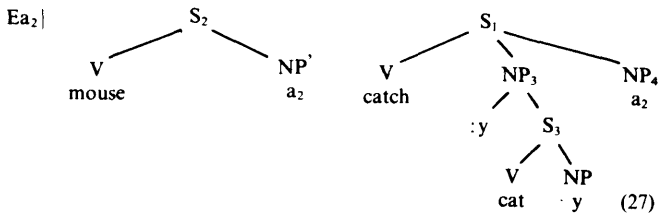
The cat caught a mouse. (26a)



(26b)

In (26b) *an* is the singular existential quantifier. SAS have generalized quantifiers, that is, binary higher order predicates over plural terms, NP_1 and NP_2. NP_1 is the matrix term (M-term) representing the M-set. NP_2 is the Q-term and represents the Q-set (the set quantified over). With existential quantification, the M-set and the Q-set are of the same order, the quantifier assigning them a nonempty intersection. (26b) is thus to be read as 'The intersection of the set of things caught by the cat and the set of mice contains at least one element' (for the grammatical treatment of quantifiers, see Seuren 1984).

Incrementally, quantifiers are 'technical' predicates, i.e., they contain a special instruction. Existential quantifiers set up a new address, of the same type as the variable under the ^-operator in NP_1 and NP_2. The new address is fitted out with the predications S_2 and S_1, in that order, with the bound variable replaced by the address variable:



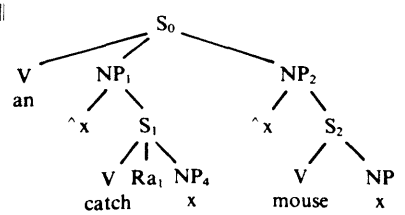
(27)

Given '$Ea_1|cat(a_1)$' in D, NP_3 is replaced by 'Ra_1' according to (d) above, resulting in:

$Ea_2|mouse(a_2), catch(Ra_1, a_2)$ (28)

i(NP_3) still has to be carried out. $d(NP_3) = Ea_1|cat(a_1)$, which is closed. NP_3 is replaced by 'Ra_1' and the entire tree (26b), with 'Ra_1' for NP_3, is added to Ea_1:

$Ea_1|cat(a_1)\|$ (29)



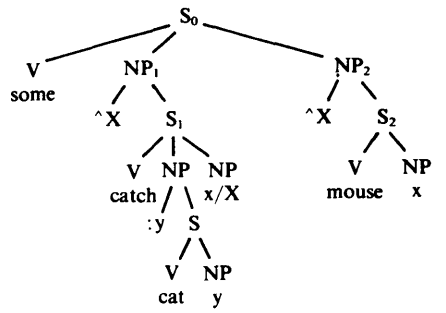
Now S_0 must be incremented again. This gives:

$Ea_1|cat(a_1) \| Ea_2|mouse(a_2), catch(Ra_1, a_2)$ (30)

or 'the cat is one of the set of individuals that caught a mouse.' (28) and (30) together are the incremental result of (26a).

Now consider (31) (the predicate *some* represents the plural existential quantifier):

The cat caught (some) mice. (31a)



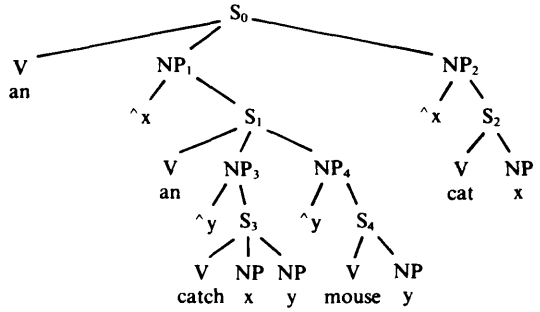
(31b)

This runs parallel to (26), with the extra proviso that a 1st-order predication 'Pred(x)' under the operators '^X' or ':X' corresponds to '::$Pred(a_n)$.' For intrinsically 2nd-order predicates (e.g., *disperse*, *be numerous*) 'Pred(X)' is incremented as '$Pred(A_n)$.' Since group readings automatically type-raise the predicates concerned (provided they are not semantically restricted to individuals), S_1 in (31b) has the alternative variables x and X. With x, I produces (32a), i.e., 'the cat caught individual mice'; with X (32b) or: 'the cat caught a set of mice':

(a) $Ea_1|cat(a_1) \| EA_2|::mouse(a_2), ::catch(Ra_1, a_2)$ (32)
$EA_2|::mouse(a_2), ::catch(Ra_1, a_2)$
(b) $Ea_1|cat(a_1) \| EA_2|::mouse(a_2), catch(Ra_1, A_2)$
$EA_2|::mouse(a_2), catch(Ra_1, A_2)$

Double existential quantification is treated analogously. (33b) yields (33c):

A cat caught a mouse. (33a)



(33b)

$Ea_1|cat(a_1), Ea_2|mouse(a_2), catch(a_1, a_2)$ (33c)

To account for the fact that a text may continue with a definite description like *the mouse*, a process of 'inferential bridging' must be assumed, deriving from (33c) an independent open address:

$Ea_2|mouse(a_2), Ea_1|cat(a_1), catch(a_1, a_2)$ (34)

The setting up of (34) is subject to constraints which are not clear at the moment, given the insufficiency of available knowledge of plurals. The problem appears, for example, in sentences like:

Few cats chased many mice. (35)

I will, in the distributive reading, correctly yield a new plural cat-address:

$EA_1|::cat(a_1), few(A_1), EA_2|::mouse(a_2), many(A_2),$ (36)
$::chase(a_1, a_2)$

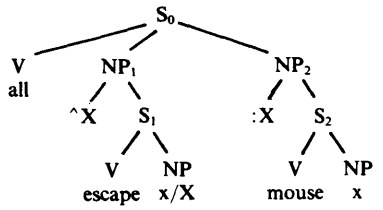
but must be prevented from incorrectly yielding also a new distributive address:

$EA_2|::mouse(a_2), many(A_2), EA_1|::cat(a_1), few(A_1), ::chase(a_1, a_2)$ (37)

which corresponds to *Many mice were chased by few cats.* This sentence is equivalent to (35) only in the strictly collective reading according to which a small set of cats chased a large set of mice. Given the lack of clarity in this matter, the process of inferential bridging is left out of account here.

Now consider universal quantification, as in:

All (the) mice escaped. (38a)



(38b)

With lower case x, that is, in the distributive reading, (38b) reads as 'the discourse-defined set of mice $M \in P_{pl}(\{x|escape(x)\})$.' Capital X represents the group reading 'the discourse-defined set of mice M is one of the groups that escaped' (compare (16a, b) above). Let D contain '$EA_1|::mouse(a_1)$.' Universal quantifiers take the plural address denoted by NP_2, in this case EA_1, ordering the addition of S_0, with NP_2 replaced by RA_1. NP_1 is rendered as '[::escape(x)],' or as '[escape(X)].' The result is the distributive (39a) (=(16a)) or the group-reading (39b):

$EA_1|::mouse(a_1) \| all([::escape(x)], RA_1)$ (39a)

$EA_1|::mouse(a_1) \| all([escape(X)], RA_1)$ (39b)

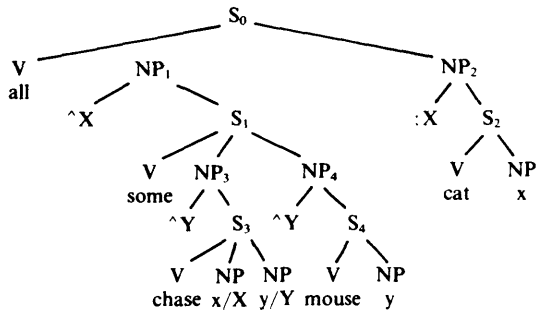
Note that this machinery makes it possible to distinguish in a natural way between *every* and *all*. The difference resides in their subcategorization conditions. *Every*, in particular, requires distribution in NP_1, so that, for example, **Every mouse dispersed* becomes ungrammatical and uninterpretable, and *Every mouse escaped* can be interpreted only as (40a), and not as (40b):

$EA_1|::mouse(a_1) \| every([::escape(x)], RA_1)$ (40a)

$!\,EA_1|::mouse(a_1) \| every([escape(X)], RA_1)$ (40b)

Now consider (41), with D containing '$EA_1|::cat(a_1)$':

All (the) cats chased (some) mice. (41a)



(41b)

Application of the above principles gives four possible incrementations:

$EA_1|::cat(a_1) \| all([::EA_2|::mouse(a_2), ::chase(x, a_2)], RA_1)$ (42a)

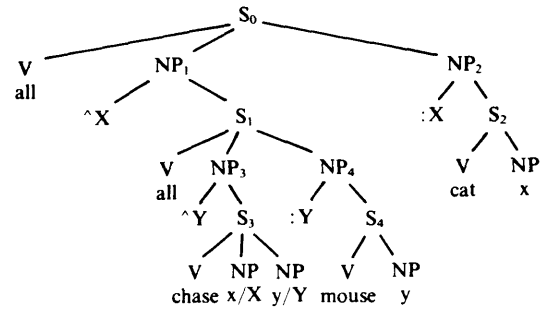
$EA_1|::cat(a_1) \| all([::EA_2|::mouse(a_2), chase(x, A_2)], RA_1)$ (42b)

$EA_1|::cat(a_1) \| all([EA_2|::mouse(a_2), ::chase(X, a_2)], RA_1)$ (42c)

$EA_1|::cat(a_1) \| all([EA_2|::mouse(a_2), chase(X, A_2)], RA_1)$ (42d)

Finally the following is discussed:

All (the) cats chased all (the) mice. (43a)



(43b)

As the reader will be able to work out, the result is fourfold, each reading specifying group or distributive readings for the set of cats (RA_1) and the set of mice (RA_2):

$EA_1|::cat(a_1) \| all([_x::all([_y::chase(x, y)], RA_2)], RA_1)$ (44a)
$EA_2|::mouse(a_2) \| all([_y::all([_x::chase(x, y)], RA_1)], RA_2)$

$EA_1|::cat(a_1) \| all([_X all([_y::chase(X, y)], RA_2)], RA_1)$ (44b)
$EA_2|::mouse(a_2) \| all([_y::all([_X chase(X, y)], RA_1)], RA_2)$

$EA_1|::cat(a_1) \| all([_x::all([_Y chase(x, Y)], RA_2)], RA_1)$ (44c)
$EA_2|::mouse(a_2) \| all([_Y all([_x::chase(x, Y)], RA_1)], RA_2)$

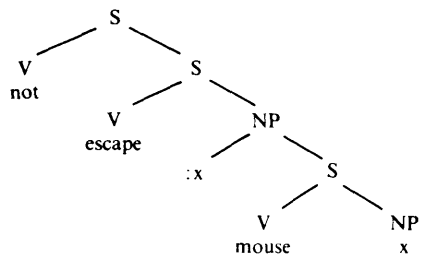
$EA_1|::cat(a_1) \| all([_X all([_Y chase(X, Y)], RA_2)], RA_1)$ (44d)
$EA_2|::mouse(a_2) \| all([_Y all([_X chase(X, Y)], RA_1)], RA_2)$

Analogous treatments can be provided for noncanonical quantifiers like *most*, *half*, *many*, *few*, or the default quantifier occurring in generics.

1.3 Negation

In SA, negation is an abstract predicate, like the quantifiers. Its subject-S is what is normally called its scope. Thus, (45a) has the SA (45b) (disregarding tense):

The mouse did not escape. (45a)



(45b)

In D-structures, *not* is represented as '*' preceding the predicate just below it in SA:

$Ea_1|mouse(a_1) \| {}^*escape(Ra_1)$ (46)

The discourse-semantic function of negation is an instruction preventing the addition of the predication immediately following it to the address in question. For the instruction to work, the nonnegated predication must be normally incrementable. I takes the subject-S of *not* in the SA-tree and processes it first in the normal way, i.e., without negation.

Subsequently, *not* causes an asterisk to be placed, in principle (but see below for the type-sensitivity of negation), before the predication resulting from its subject-S. In the case at hand, there must, therefore, be an appropriate mouse-address available in D. In (46), '*escape(Ra_1)' is to be read as 'the mouse belongs to the class of those that did not escape.' Generally, * is presupposition-preserving, and differs, therefore, from the negation in classical logic (see *Presupposition*).

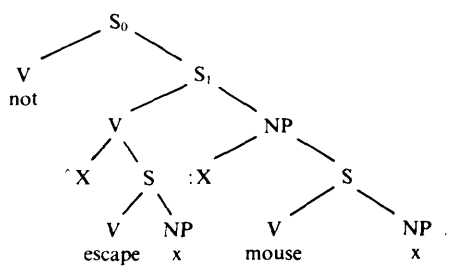
Since negation is typed (t, t) it requires an input typed t or (X, t). Open addresses can, therefore, be negated:

$$*Ea_1 | \text{mouse}(a_1) \quad (47)$$

(*There is/was no mouse*). The asterisk prevents the addition of any address of the form 'Ea_n|mouse(a_n)' to D. But negated open addresses cannot be closed, due to the type-shift after closure. In (47), the head 'Ea_1' is typed ((e, t), t), but in (46) it has been retyped ((e, t), e), no longer providing an input for the negation function (t, t). The machinery thus automatically accounts for the fact that no definite anaphoric reference is possible to a negated existential.

Negation is, apparently, type-sensitive. That is, it checks the basic type of the 'nuclear' predicate of its subject-S in the SA-tree and ends up in D as '*' just before the predication of the right type. Thus, in (48b), the verb *escape* is the nuclear predicate of the main V of S_1, and is typed (e, t). Therefore, '*' ends up in (48c) before *escape*, and not before '[: : escape(x)],' which is typed ((e,t),t). This accounts for the fact that (48a) entails that none of the mice escaped, not just that not all mice did:

The mice did not escape. (48a)



(48b)

$$EA_1 | : : \text{mouse}(a_1) | [: : *\text{escape}(x)](RA_1) \quad (48c)$$

In general, '*' will never immediately precede the : :-operator in any address.

Likewise for universal quantification. The negation of (38), or:

Not all (the) mice escaped. (49)

is incremented, analogously to (39), as either (50a) or (50b):

$$EA_1 | : : \text{mouse}(a_1) || *\text{all}([: : \text{escape}(x)], RA_1) \quad (50a)$$

$$EA_1 | : : \text{mouse}(a_1) || *\text{all}([\text{escape}(X)], RA_1) \quad (50b)$$

The asterisk stands before *all*, the nuclear predicate of the negated S, whose basic type is ((e, t), t). Application of these principles yields, correctly, ($51b_{1,2}$) for (51a), interpreted with large scope for *all*:

All (the) mice did not escape. (51a)

$$EA_1 | : : \text{mouse}(a_1) || \text{all}([: : *\text{escape}(x)], RA_1) \quad (51b_1)$$

$$EA_1 | : : \text{mouse}(a_1) || \text{all}([*\text{escape}(X)], RA_1) \quad (51b_2)$$

A sentence like (52a) is incremented as (52b), and (53a) as (53b):

No mice escaped. (52a)

$$*EA_1 | : : \text{mouse}(a_1), : : \text{escape}(a_1) \quad (52b)$$

Some mice did not escape (53a)

$$EA_1 | : : \text{mouse}(a_1), : : *\text{escape}(a_1) \quad (53b)$$

2. The Theory of Domains and Subdomains

2.1 The Sequentiality Criterion

The incrementation of a sequence of sentences in D is subject to certain conditions, some of which form the 'sequentiality criterion' (SC) (also discussed in *Presupposition*). SC does not apply to actual texts, which are hardly ever fully sequential (a fully sequential text would be unbearably verbose), but to Ds. Besides SC, there are other structuring principles for texts and Ds, which will not be discussed here. Foremost among these is the principle of 'topic-comment modulation,' based on the notion that each new sentence in a discourse is meant as an answer to an, often implicit, question. (Van Kuppevelt 1991; see *Topic and Comment*).

A general condition on the incrementation of any sentence A is that all predicates occurring in A have 'cognitive backing,' that is, the relations they enter into in the D under construction form, or fit into, a 'scenario' that makes functional sense. This condition is particularly relevant for the complex of possessive predicates (*have, with, of*, genitives, etc.). The semantic description of these predicates contains an open parameter whose value is to be somehow retrieved from general or particular background knowledge (see *Psychological Semantics*). For example, to say of a hotel that all its rooms *have* a shower implies truth-conditionally that there is a specific shower for (in or near) each room. But to say of a faculty that all its students *have* a supervisor does not require such one-to-one mapping. This is because the predicate *have* refers the user to his or her background knowledge, which tells him or her what it is for a hotel room to have a shower, and what it is for a student to have a supervisor. It will, normally, fail to tell the user, for example, what it is for a room to have an engine. Hence the fact that a sequence of sentences like *She entered a room, but she could not get its engine started* will normally be uninterpretable. Similar conditions hold for compounds. Unless a special discourse has provided one, there is no cognitive backing, for example, for a compound like *wheel fire*. Compounds depend on background knowledge for a specification of the relation between their constituent parts. Needless to say, the condition of cognitive backing has so far eluded any attempt at formal treatment.

More formally manageable are the specific conditions of SC. SC implies that the incrementation of a conjunction $A \wedge B$, that is, i($A \wedge B$), consists, at least in principle, simply of i(A) followed by i(B): D then contains A + B. By definition, the set of verification domains V ('worlds') in which D is true, or /D/, is the intersection of the set of Vs in which A is true, or /A/, and the set of Vs in which B is true, i.e., /B/. Generally, $/D/ = \bigcap_S$ for all $S \in D$. If and how *and* is to be represented in D has so far remained an open question. Where applicable, the comma that links predications may be taken to represent it. Further research will

have to reveal how the semantic differences with other conjunctive operators, such as *but* or *for*, should be expressed. In any case, the structural conditions for negation over *and* are not fulfilled. Likewise, a conjunction cannot carry presuppositions: presuppositions are properties of sentences that are incrementable as a single unit (see *Presupposition*).

Moreover, SC requires that any newly added incrementation restricts the set of Vs in which D is true: for any i(A) to D, $/D+A/ \subset /D/$. This is the 'informativeness condition' for Ds. It implies, for example, that D does not accept i(A or B) if i(A), or i(B), is already in D. And 'possible(A)' is not acceptable if $i(A) \in D$ (though incrementation is still allowed in intensional subdomains introduced by *therefore* and the like). Yet, no new i(A) may restrict /D/ to such an extent that $/D/ = \varnothing$. In other words, SC requires that no incrementation i(A) is to result in the empty set of possible Vs for which D is true. That is, D must be consistent. We therefore speak of the 'consistency condition.' The consistency condition is a much heavier constraint than the informativeness condition, which does not preclude interpretability the way the consistency condition does.

Finally, SC requires that no sentence be incremented unless all its presuppositions have been incremented first. This does not mean that all the presuppositions of a sentence must be actually uttered in the text that is produced (that would be unbearably verbose). Since presuppositions are structurally retrievable from their carrier sentences, I can retrieve them and supply them by accommodation (see *Accommodation and Presupposition*). Accommodation of a presupposition is, of course, subject to the conditions of cognitive backing, informativeness, and consistency.

Interestingly, at any stage of development of a D there is a set of sentences of the language that simply cannot be processed in D, on account of the consistency condition. (The condition of cognitive backing can be satisfied ad hoc by providing suitable background information, and the informativeness condition does not, strictly speaking, affect interpretability.) This set, the 'nonlanguage of D' or NL(D), consists of the sentences whose presuppositions have been denied in D. The radical negation operator NOT (i.e., *not* with heavy accent) is used to say of a sentence that it belongs to NL(D). The normal (minimal) negation as in 'not-A,' incremented as '*,' yields truth for the complement of /A/ in /D/ after all A's presuppositions have been accommodated, not for the complement of /A/ in the universe of all possible Vs, as classical negation does. A gapped logic thus seems required to describe the logical properties of sentences incremented in Ds. Or else, only those sentences that are interpretable in any given D must be considered valid objects for logical treatment, which means that the logic must work with an ever changing universe of discourse (see *Presupposition* for a fuller discussion).

2.2 *Alternative Incrementations and Subdomains*

'Alternative incrementations' have not been discussed so far. They result mainly from the operators *or* and *if* and epistemic possibility. Disjunctions, that is, sentences of the form 'A or B' are incremented as consisting of the alternatives 'A' and 'not-A and B,' or: 'A/[not-A and B].' Likewise, 'not-A or B' is incremented as 'not-A/[A and B].' The truth-condition is that at least one of the alternatives be true. A sentence like *John is happy or he is ignorant* is incremented as shown in (54) (with '"John"' for 'be called "John"'):

$$Ea_1 | \text{"John"}(a_1) \,\|\, [\text{happy}(Ra_1)/{}^*\text{happy}(Ra_1), \text{ignorant}(Ra_1)] \qquad (54)$$

This accounts, in principle, for the much debated 'exclusive' character of natural language *or*, since the alternatives in (54) cannot both be true.

It also accounts for 'donkey' anaphora in a disjunction, as in (55a, b):

Either John bought no car, or it is white. (55a)

$$Ea_1 | \text{"John"}(a_1) \,\|\, [{}^*Ea_2 | \text{car}(a_2), \text{buy}(Ra_1, a_2)/Ea_2 | \text{car}(a_2), \text{buy}\ (Ra_1, a_2) \,\|\, \text{white}(Ra_2)] \qquad (55b)$$

Since, in the second alternative, the address Ea_2 has been closed, 'Ra_2' stands for the car that John bought if he bought one. The predication after closure reads as 'either John bought no car or he did buy a car and that car is white.'

Implications, another source of donkey anaphora, are considered to involve at least the following treatment. A sentence of the form 'if A then B' is incremented as 'not-A or B,' that is, 'not-A/[A and B].' Analogously, 'if not-A then B' becomes 'A/[not-A and B].' Thus i(56) involves at least the structure of (55b):

If John bought a car it is white. (56)

DRT, mentioned above, has essentially the same approach to donkey anaphora. It also gives up the premise that an anaphoric pronoun in a true extensional sentence must be analyzed as either a bound variable or a term with fixed reference. The intermediary stage of discourse domains allows for the view that anaphora, and, for that matter, reference, is not a direct relation between a sentence and a verification domain V, but is mediated by a cognitive discourse structure.

One must note, however, that though this affords some breathing space with regard to the donkey sentences problem, it does not solve it. This appears from the fact that predications involving at least one definite term must still have a truth-value even when the definite term lacks reference in V, if the disjunction and implication operators are to remain truth-functional. Assignment of simple falsity will not do, as the negation operator would then yield truth, which would be wrong in any theory taking account of presuppositions as discourse phenomena, such as DSX. It seems, therefore, that a DSX solution to this problem necessitates (a) the acceptance of intensional objects as reference values (see Seuren 1985: 472–76), and (b) the acceptance of a third truth-value ('radical falsity'). If these conclusions hold, then the proper logic for DSX is trivalent, and also gapped if the universe of discourse is to remain constant. DRT will face the same dilemma when it gets down to presuppositions.

It is in no way implied, of course, that the analysis presented here provides an answer to all questions arising in connection with disjunctions and implications. It seems, for example, that an implication 'if A then B' is appropriate only if the possibility of A has been raised in previous discourse. It also seems that the addition of B after A in the second alternative is normally qualified by some modality

('must,' 'will'). Nothing more than a partial framework can be given here.

Subdomains represent what is usually called 'intensional contexts' (see *Intensionality*). They are a special kind of address. Like ordinary addresses, they contain predications and can be closed. They also have domain properties, in particular the property of being able to contain addresses, predications, and instructions, which are limited to the subdomain in question. A double notation is therefore used: they are represented both as a special kind of address and as a (subscripted) domain. A sentence like (57a) is incremented as (57b), which contains both the domain-address D_1 and the specification of what D_1 contains. The main domain (truth domain) D is assumed to contain the address '$Ea_7|$"Mars"(a_7), planet(a_7),' which is then closed and enriched as in (57c). Note that in D_1 Ea_7 is immediately closed and given the predication 'inhabited(Ra_7),' indicating that Ea_7 has been 'borrowed' from D so that the object referred to is taken to exist in the real world (so-called 'transparent reference').

It is possible that Mars is inhabited. (57a)

$$D_1 \mid \text{possible}(D_1) \qquad (57b)$$
$$D_1\ \overline{Ea_7 \parallel \text{inhabited}(Ra_7)}$$

$$E_7 \mid \text{"Mars"}(a_7),\ \text{planet}(a_7) \parallel \text{possible}(RD_1) \qquad (57c)$$

Opaque reference occurs in, for example, (58a), incremented as (58b), with **D** containing Ea_1 for *John*, closed and expanded as in (58c):

John believes that there is a planet Minerva and that this planet is inhabited (58a)

$$ED_1 \mid \text{believe}(Ra_1, D_1) \qquad (58b)$$
$$D_1\ \overline{Ea_1 \mid \text{"Minerva"}(a_1),\ \text{planet}(a_1) \parallel \text{inhabited}(Ra_1)}$$

$$Ea_1 \mid \text{"John"}(a_1) \parallel \text{believe}(Ra_1, D_1) \qquad (58c)$$

Now $Ea_1^{p_1}$ is properly opened and closed within D_1, so that the existence of this planet is limited to the world of John's beliefs. The definite NP *this planet* now refers opaquely.

A few general principles hold for subdomains. First, addresses from the truth domain 'percolate' downwards into subdomains, as shown in (57), where the address Ra_7 is taken from D. This downward percolation is stopped only if the subdomain in question explicitly negates the existence of the referent in question. Then, presuppositions of clauses incremented in subdomains must in any case be introduced into the subdomain in question (according to SC), but they will, by way of a default procedure, 'climb' up into higher domains and into D unless stopped either by their explicit negation or by lack of cognitive backing. This process is called 'projection' (see *Projection Problem*). Both downward percolation of addresses and upward projection of presuppositions serve the functional purpose of ensuring maximal unity and coherence in the whole D-structure.

Anaphora may delve into subdomains. The predicate *exist*, for example, being intensional with regard to its subject term (see *Existence Predicate* (*Discourse Semantics*)), naturally makes its subject term look for an address in some intensional subdomain. Its incremental result is that this address is given a place in the truth domain. Thus in:

Marion believes that she has a brother, but he does not exist. (59)

the pronoun *he* closes the brother-address set up in Marion's belief-domain, and the negated existential statement prevents it from being added to the truth domain. In this respect, anaphora contrasts with presuppositions, which can never be retrieved from intensional subdomains. The sentence:

!Marion is under the illusion that she has a brother. He lives in Texas. (60)

is incoherent because the presupposition of the second sentence ('Marion has a brother') is not fulfilled in the truth domain, only in the subdomain of Marion's illusions, which, in virtue of the meaning of *illusion*, is per se barred from the truth domain. Even so, the pronoun, *he* is clearly anaphoric to Marion's thought-up brother. Heim's (1982) suggestion that presupposition is really a special form of anaphora thus seems hard to uphold.

A further principle of domain economy concerns the structuring of alternatives. When a sequence of alternative domain splits occurs, new alternatives will latch on to old ones to the extent that SC allows for it. Thus, a sequence like:

If John is not in his office he is with his girlfriend. And if his wife finds out he is in for trouble. (61)

splits up D into four hierarchically ordered alternatives: one in which John is in his office and one in which he is not but with his girlfriend, in which case either his wife does not find out or she does and he is in for trouble.

In addition to the cognitive aspects of sentence comprehension, a proper elaboration of the machinery outlined should account also for the logical and intensional properties of sentences, as well as for the phenomena that have been observed in connection with presuppositions.

See also: Accommodation and Presupposition; Presupposition; Projection Problem; Topic and Comment.

Bibliography

Fauconnier G 1985 *Mental Spaces: Aspects of Meaning Construction in Natural Language*. MIT Press, Cambridge, MA

Gazdar G 1979 *Pragmatics, Implicature, Presupposition, and Logical Form*. Academic Press, New York

Geach P 1962 *Reference and Generality: An Examination of Some Medieval and Modern Theories*. Cornell University Press, Ithaca, NY

Heim I 1982 The semantics of definite and indefinite noun phrases (Doctoral dissertation, University of Massachusetts)

Heim I 1990 E-type pronouns and donkey anaphora. *Linguistics and Philosophy* **13**: 137–77

Isard S 1975 Changing the context. In: Keenan E (ed.) *Formal Semantics of Natural Language*. Cambridge University Press, Cambridge

Kamp H 1981 A theory of truth and semantic representation. In: Groenendijk J, Janssen T, Stokhof M (eds.) *Formal Methods in the Study of Language*, vol. 1. Mathematisch Centrum, Amsterdam

Kamp H, Reyle U in press *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht

Russell B 1905 On denoting. *Mind* **14**: 479–93

Seuren P A M 1972 Taaluniversalia in de transformationele grammatika. *Leuvense Bijdragen* **61**: 311–70

Seuren P A M 1975 *Tussen taal en denken: Een bijdrage tot de empirische funderingen van de semantiek*. Oosthoek, Scheltema, Holkema, Utrecht

Seuren P A M 1984 Operator Lowering. *Linguistics* **22**: 573–627

Seuren P A M 1985 *Discourse Semantics.* Blackwell, Oxford
Strawson P F On referring. *Mind* **59**: 320–44
Strawson P F 1952 *Introduction to Logical Theory.* Methuen, London
Van Kuppevelt J 1991 Topic en comment: Expliciete en impliciete vraagstelling in discourse (Doctoral dissertation, Nijmegen University)

P. A. M. Seuren