An event semantics for the Theta System

Alexis Dimitriadis

Utrecht institute of Linguistics OTS

March 16, 2011

This work was inspired by Tanya Reinhart's seminars on the Theta System in 2001 and 2002. Earlier versions of this text were circulated in 2002 and 2004 (Dimitriadis 2004), and portions have been presented at the 26th GLOW Colloquium in Lund, Sweden, at the Brown University Workshop on Direct Compositionality (both in 2003), and to audiences at the Utrecht institute of Linguistics and elsewhere.

The Theta System was still evolving during this time period, and some of the insights developed below (in particular, the adoption of event semantics) have become part of the Theta System as presented in Reinhart and Siloni (2005). While I have revised this text for publication and reorganized the order of presentation, I have kept the focus of the original work, which addresses the Theta System as presented in Reinhart (2000, 2002). Some topics have been augmented with references to later literature, and the terminology has been adapted to reflect the latest presentations of the Theta System (e.g., "decausativization" instead of "expletivization"). A new, substantial appendix proves the claim that certain arity (valence changing) operations cannot be performed on model-theoretic functions, and identifies the restricted conditions under which these manipulations become possible.

1 Introduction

The theory of the Theta System (Reinhart 2000, 2002, forthcoming) gives a compositional account of verbal argument structure and argument structure alternations. Although the Theta System makes concrete predictions with respect to argument projection and syntax, Reinhart did not provide a definition of the semantics, and semantic operations, associated with the system's component parts. In this paper I develop a semantic implementation of the system's primitives, based on a straightforward

I am grateful to Tanya Reinhart, Marijana Marelj, Tali Siloni, Maribel Romero, Danny Fox and Roger Schwarzschild for suggestions, questions, answers and discussion that have contributed to the direction and substance of the work presented here. Thanks also to Tali Siloni and two anonymous reviewers for their many suggestions on the form and content of this version of the text.

embedding of the Theta System in the *event semantics* of Davidson (1967), Parsons (1990). While it had not been my intent to extend or "explain" the Theta System, the design proposed in these pages turns out to have interesting empirical consequences; these are explored in later part of the paper.

The Theta System does not consider the order of thematic arguments to be encoded in the lexical entry; it is determined after the operation of general marking and arity operations (which manipulate the argument inventory), according to dedicated rules of the computational system (CS), the *CS merging instructions.* But syntactic constituency, as well as the lambda forms standardly used in formal semantics, involve argument positions that are arranged in a particular order. In order to end up with the right kind of semantic object, there must be a *change of representation* at some point between the lexical entry and the form of the sentence at the LF interface. Before the change-over, which I will refer to as *assembly,* argument structure information is represented as a complex structure which we are at liberty to define as it suits us; afterwards, we must deal exclusively with the entities standardly employed by current theories of grammar: functions in a model-theoretic logic for the semantics, and aggregates of syntactic features for the syntax. I will show that it is best to locate this change just before lexical items enter syntactic computation (e.g., at the point of insertion into the numeration).

In the model I propose, the thematic role inventory of a stored lexical entry is a set of theta clusters, with no explicit order. Word-building in the lexicon involves copying the contents of the stored entry into a "working structure" (WS), which is then manipulated by arity operations and enriched with calculated properties such as the Accusative case feature and, eventually, the order of argument projection. Just before lexical items enter syntactic derivation, the assembly step takes place. It removes theta clusters one by one from the set of arguments of the WS, in the order dictated by the CS merging instructions, and conjoins them into a single model-theoretic function that expresses the denotation of the verb. The WS is converted into a simpler syntactic object, consisting of syntactic features and a single function of event semantics encoding verb meaning. After this point, thematic arity operations must operate on the assembled object; their effect on the verb's meaning must be expressible as manipulations of the model-theoretic function representing the verb's denotation.

This architecture, adopted as a natural solution to the practical problem of designing a semantics of the Theta System, accounts in a natural way for certain known distributional asymmetries of arity operations. In particular, the Theta System recognizes two domains of application for arity operations: the lexicon and the (morpho)syntax. While some arity operations, such as reflexivization, take place in

the lexicon in some languages and in the syntax in others, Reinhart and Siloni (2005) discovered that other arity operations are cross-linguistically restricted to the lexicon. Prior to the work reported here, there was no comprehensive explanation of why some arity operations can apply in either domain but others cannot, or of which ones are so restricted.[1]

In formalizing both types of arity operations within the semantic framework I will propose below, it turns out that the arity operations that apply in the syntax are exactly those that can be expressed as basic manipulations of the corresponding semantic denotation: Specifically, an operator on verb denotations expressed as type-logical functions can existentially close off a role or identify two roles; but it cannot entirely delete the first (or n-th) theta role from the truth conditions of the verb's denotation, nor replace it with another. On the empirical side, the arity operations which Reinhart and Siloni found to be restricted to the lexicon require outright deletion of a theta role and/or substitution of one thematic role for another (reduction, causativization), while those permitted in either domain involve existential closure of an argument position, or identification of two theta roles (saturation, arbitrarization, reflexivization). I will argue that this correlation *explains* the distribution of arity operations: An operation can take place in the syntax only if its effect can be expressed as a manipulation of the semantic object present in syntactic derivation. A different, more articulated linguistic object encodes verb semantics in the lexicon, which can undergo the full complement of arity operations.

While it could be objected that we are dealing with accidental consequences of a postulated system, our findings derive from limitations on what is possible during syntactic derivation; and the encoding I have assumed for the syntactic component is simply the usual view of word meaning as functions in a logical calculus. The complex linguistic object I adopt for the lexicon has no interesting properties: It is motivated by the Theta System's model of arity operations and thematic role ordering (that is, for reasons entirely independent of the distributional differences between the lexical and syntactic components), and it simply makes it possible for these operations to be carried out. The interesting finding is that this custom object is not needed beyond the point of lexical insertion: Of the arity operations posited by the Theta System, those that cannot be expressed as manipulations of standard denotation functions are exactly the ones that are unattested in the syntactic component. The semantics of arity operations that do take place in the syntax, in the account of the Theta System, can

[1] Siloni (2002), Reinhart and Siloni (2005) adopt a related account, discussed in section 5.1.

all be expressed as simple operations on model-theoretic denotation functions. This pattern of distribution supports the Theta System's division of arity manipulations into two components, and confirms the view that the expressive power of the standard semantic framework is approximately right for a semantics of natural language.

For reasons of space, I do not provide a detailed summary of the Theta System, nor (especially) of the evidence for its claims. I do discuss many aspects of the theory as they relate to the claims made below. While I have attempted to make the presentation self-sufficient, readers who are entirely unfamiliar with the Theta System may wish to consult the introduction to this volume, or Reinhart (2002, forthcoming) and Reinhart and Siloni (2005) for the full details and motivation.

The paper is organized as follows: In the next section, I review the general organization of the Theta System and some of its more relevant constituent parts. Section 3 discusses the embedding of event semantics in the framework of the Theta System and defends the conclusion that there is a change of representation between the lexicon, where theta roles have no intrinsic ordering, and the eventual semantic interpretation. While the initial impetus for this decision is the transition from unordered to ordered theta roles, section 3.4 discusses an immediate benefit (and additional motivation): Several of the Theta System's arity operations cannot be carried out on simple denotation functions. Section 4 presents an explicit derivational model for the operations of the Theta System. Finally, section 5 takes up the issue of the lexicon–syntax division, and the correspondence between the observed distribution of arity operations in the two components and the predictions of the formal model proposed here. The formal results claimed in section 3.4 are proved in a separate appendix (section 7).

2 The Theta System

The Theta System framework (Reinhart 2000, 2002, forthcoming, Reinhart and Siloni 2005) provides a compositional account of verbal argument structure and argument structure alternations. The theta system is "the central system of the systems of *concepts*", and is responsible for putting concepts into a form that is legible by the computational system (CS). It includes, inter alia, lexical entries (encoding concepts and theta role specifications), a set of operations on argument structure and marking procedures, and other general rules for the interaction of the above. Reinhart focuses on accounting for general, cross-linguistic patterns in the argument structure of verbs: in particular, on argument ordering

and co-occurrence restrictions, and on allowable structural alternations like the causative–unaccusative alternation, shown in (1).

(1) a. John opened the door.

b. The door opened.

The Theta System derives a good deal of what is currently known about these phenomena from a system of primitive structures and operations, subject to general conditions, that derive verb entries from coded concepts in the *mental lexicon.* It is understood that stored concepts are basic and non-redundant; variant argument realizations are generated from the same underlying concept by application of various operations on argument structure, such as passivization and causativization. The stored verbal concepts do not specify the order or manner of projection (internal or external) of verbal arguments; these are determined by general principles, the *CS merging instructions,* which are sensitive to the application of argument structure operations.

I will make a terminological distinction, not clearly made in Reinhart's writings, between *argument structure operations* and *arity operations.* The former are grammatical devices that may vary cross-linguistically in their details: the passive, the antipassive, and reflexive verb formation in various languages are examples.[2] These operations employ a very restricted set of universal *arity operations,* each of which performs a particular manipulation of theta grids. Arity operations include *decausativization* (deletion, also known as *reduction,* of an external theta role expressing Cause), *saturation* (existential closure of a role), and *causativization* (which transforms a verb's theta grid, adding an agent and suppressing the causal status of the original agent or cause).[3]

For example, the passive is an argument structure operation realized slightly differently in different languages: intransitive verbs can be passivized in Dutch, but not in English, and so on. But the passive of every language utilizes the arity operation of saturation (existential closure). I will assume that the properties of arity operations are invariant; cross-linguistic differences are associated with the language-particular passive constructions, not with saturation itself.

[2]Argument structure operations, as defined here, are also known as *valence changing* or *valence adjusting* operations. Baker (1988) calls them *grammatical function changing processes* (or *GF-changing processes*).

[3] Note that some of these terms have different (and sometimes varying) meanings in the literature; in this text they are used with the meaning assigned to them in the Theta System.

According to the Theta System, arity operations take place in two domains, the *computational lexicon* and the *syntax.* There is some redundancy in the system: Reinhart and Siloni (2005) found that several arity operations (e.g., reflexivization) apply either in the (computational) lexicon or in the syntax, depending on the language. Others, such as causativization, are restricted to the lexicon. As mentioned already, the present paper proposes a partial explanation for this state of affairs.

2.1 Theta clusters

The Theta System represents thematic roles as combinations of the features [±c], [±m] (standing for *cause change* and *mental state,* respectively). For example, the *Agent* role corresponds to the cluster [+c+m]. The system is actually ternary, not binary, since a feature may take a plus value, a minus value, or be absent; there are thus nine possible feature clusters, including the empty cluster which cannot be realized directly.[4]

As with any system of theta roles, determining the feature clusters (theta clusters) in the lexical entry of particular verbs is not an exact science. The appropriate cluster must be deduced from a combination of the argument's semantics and the verb's combinatorial properties. The feature clusters corresponding to the major theta roles (such as Agent) can generally be assigned on the basis of their semantics, while the presence of other roles like [−m] (subject matter or "target of emotion") must often be inferred from a verb's argument structure alternations, in part relying on the predictions of the theory. (For a list of the theta clusters and their correspondences to traditional theta roles, see the introduction to this volume).

2.2 Argument projection and linearization

As already mentioned, the lexical entries of verbs do not specify argument order or internal vs. external projection. These are computed on the basis of a verb's inventory of theta clusters, according to the principles of the *CS merging instructions.*[5] Arguments are not ordered according to a universal, explicit

[4]The empty cluster was presumed to be non-occurring in earlier formulations of the Theta System. It is introduced by Marelj (2004) as the underlying theta role that gives rise to middles.

[5] The Theta System adopts a version of the Universal Alignment Hypothesis (UAH) (Perlmutter and Postal 1984:97), maintaining that the ordering and internal vs. external projection of the theta roles of any given verb is subject to universal constraints. Unlike some stronger claims (cf. Baker's (1988:46–48) Uniformity of Theta Assignment Hypothesis), the UAH does not assume an invariant structural position for each theta role. See Pesetsky (1995:11ff) for a discussion and comparison of the two claims.

linear hierarchy as usually proposed (see, for example, Grimshaw 1990). Rather, the following marking and merging principles determine the manner and order of argument projection:

(2) **Lexicon marking:** Given an n-place verb entry, $n > 1$,

a. Mark an all-plus cluster ([+c], [+m] or [+c+m]) with index 1.

b. Mark an all-minus cluster ([−c], [−m] or [−c−m]) with index 2.

c. If the entry includes both an all-plus cluster and a fully specified cluster [α,−c] (i.e., [+m−c] or [−m−c]), mark the verb with the ACC feature. This indicates that the verb has Accusative case to assign.

(3) **CS merging instructions:**[6]

a. When nothing rules this out, merge externally.

b. An argument realizing a cluster marked 2 merges internally; an argument with a cluster marked 1 merges externally.

c. In accordance with instruction (a), an unmarked argument merges externally unless:

(i) some other argument is already merging externally. Or,

(ii) the verb has the ACC feature, and no other argument is available to receive Accusative case.

Application of the marking rules in (2) is explicitly restricted to verb entries with more than one argument. Verbs that are underlyingly intransitive, and do not undergo any arity operations that add an argument (see below), are not marked with projection indices. The merging instructions cause all such verbs to project their argument externally (since "nothing rules this out"). But an arity operation can detransitivize a transitive verb after it has undergone marking, which will result in unaccusative argument structure if the remaining argument carries the index 2. This process is central in Reinhart's analysis of unaccusatives. Following Chierchia (1989/2004), she derives the unaccusative variant of alternating verbs such as *break* and *open* from the transitive variant, rather than the other way around:

(4) Underlying: John opened the door.

Derived: The door opened.

[6]Part (c) is not part of Reinhart's summary of the CS merging instructions. It is added here for clarity.

This account explains why all intransitive unaccusatives have a transitive alternate, while non-alternating intransitives with just a theme argument (e.g., *glow, tremble*) are always unergative: The unaccusatives must be derived from a transitive base.[7] Theme unergatives, on the other hand, may or may not have a (derived) transitive alternate (but see Potashnik, this volume, for an alternative analysis). Readers unfamiliar with the issue, one of the centerpieces of the Theta System, are referred to Reinhart (2002, forthcoming) for details.

Note also that in the framework of the Theta System, the external argument is projected by the verb, not contributed extrinsically (e.g., by a syntactic head, as proposed by Kratzer (1996) and others).[8] By treating internal and external arguments alike, Reinhart is able to explain alternations in which the same theta role may project either internally or externally, depending on which other arguments are present. In the following sentences, *John* has the same theta role (Experiencer, [−c+m]) but projects differently: Pesetsky (1995:11ff) shows that *worry* is not unaccusative in sentence (5a), hence *John* is projected as the external argument in (5a) but is internal in (b).

(5) a. John worried.

b. The letter worried John.

The above summary of argument projection glosses over the fact that not all clusters in a theta grid need be projected; Reinhart treats certain cases as simple optionality of projection; in other cases, various conditions rule out projection of a certain cluster (e.g., an Instrument cannot be projected unless an Agent is overtly or covertly present), or certain cluster combinations may not be realized together ("indistinctness"). These conditions will not be discussed here.

2.3 Arity operations

In the Theta System, grammatical manipulations of argument structure are restricted to a small set of universal *arity operations* on theta clusters. The theory recognizes two domains in which word-building

[7] A very few unaccusatives lack a transitive base; they are considered to be derived from "frozen" transitive lexical entries, which cannot be realized outside the lexicon. (Reinhart 2002, Chierchia 1989/2004) There are also two-argument unaccusatives like *escape*, with two arguments indexed with 2. These are not problematic: Since they do undergo marking with projection indices, they can be unaccusative without being derived forms. (Reinhart 2002:17ff).

[8] For arguments against introducing the external argument extrinsically, see Horvath and Siloni (2003, forthcoming).

operations can take place: the *active lexicon,* and the *syntax.* The assumption is that operations in the syntactic component have syntax-like characteristics, while lexicon operations do not. This view of grammar, which is defended in some detail by Siloni (2002), contrasts with non-lexicalist proposals such as Distributed Morphology (Halle and Marantz 1993, Marantz 1997) which locate all word-building operations in the syntax.

Some arity operations, such as causativization, can only apply in the lexicon.[9] Others, such as reflexivization, can apply either in the lexicon or in the syntax, depending on the language. For example, reflexivization takes place in the syntax in French and Italian, but in the lexicon in Hebrew and English. These are empirical claims: causativization is defined as taking place only in the lexicon because all instances that Reinhart found are consistent with this domain of derivation. (See also Horvath and Siloni, to appear).

Reinhart did not provide a motivating rationale for every aspect of the system: making empirically correct predictions, with a minimum of over- or under-generation, was sufficient motivation. Hence these restrictions are incorporated as defining properties of the Theta System. Should they be found to be empirically false, the relevant conditions would need to be modified.

For operations that can take place in either derivational domain, Reinhart considers their distribution to be controlled by a parameter of universal grammar:

(6) **The Lex-Syn Parameter:** (Reinhart and Siloni 2005)

UG allows thematic arity operations to apply in the lexicon or in syntax.

The effect of this parameter is that only one domain of derivation (lexicon or syntax) will be utilized by any given language, and that the same domain will be utilized by all arity operations that allow parametric variation. It is not expected that some language could allow saturation to apply in one component, for example, but reflexivization in the other.[10]

Allowing arity operations to apply in two domains is less parsimonious than restricting them to a

[9]Besides the arity operation we call causativization here, there is a syntactic "causativization" construction that, as Reinhart (2002) argues, has quite different properties. The two must be viewed as unrelated constructions. While Reinhart was rather brief on this point, Horvath and Siloni (to appear) show that this second kind of causative formation, found for example in Japanese, is clearly a syntactic process: A multitude of tests demonstrate that it involves two predicates and two agents, while the causativization arity operation is uniclausal and demotes the original agent.

[10]This interpretation has turned out to be too strong; Siloni (2008) proposes that lexical reciprocals can occur even in the lexicon of languages with the parameter set to *Syntax.*

single domain, but Reinhart and Siloni argue that the distinction accounts for a large number of empirical correlations cross-linguistically. For example, diagnostics for distinguishing syntactic from lexical reflexivization include productivity (reflexivization in the syntax is productive) and reflexivization into ECM complements (possible in the syntax only, since operations in the lexicon can only target a single verb). French and Italian have reflexivization in the syntax, while in English and Hebrew it takes place in the lexicon (Reinhart and Siloni 2004). Thus the grammatical French example (7a), illustrating coreference between the matrix subject and the subject of an ECM complement, contrasts with its ungrammatical Hebrew counterpart (b).

(7) a. Jean se considère intelligent. (French)
Jean SE considers intelligent
b. * dan mitxašev 'intiligenti. (Hebrew)
Dan self-considers intelligent

While the division into lexical and syntactic arity operations is one of the core claims of the Theta System, I will not attempt to defend this view of the grammar. The question is sufficiently large that it must be deferred to Reinhart and Siloni's writings on the topic.

3 Event semantics and the Theta System

Having introduced the basics of the Theta System, we now return to the question of its semantic realization. In the terminology of the Theta System, arity operations derive new verb entries, with a different array of theta roles. Our goal is to give an account of the semantics of such derived forms, starting from the semantics of the original forms and the arity operations they undergo.

Because the Theta System presents theta roles as objects that can be manipulated and added or deleted, its model of theta clusters and their relation to the verb finds a natural counterpart in the "neo-Davidsonian" event semantics developed by Parsons (1990). In Parsons' system, a simple sentence involves explicit reference to an "eventuality" represented by a variable *e. Event types,* which are predicates over eventuality variables, correspond to the core meaning of verbs. For example, $\textit{run}(e)$ is true if e is an event of running. Thematic roles are represented as relations between this event and the corresponding participant; for example, the relation $\textit{Agent}(e, \text{John}')$ holds if John is the Agent of the event represented by *e.* An ordinary transitive sentence is represented as the conjunction of several predicates over an eventuality variable. The meaning of sentence (8a) is given in (b).

(8) a. John kicked the ball.

b. $\exists e\,\text{kick}(e) \wedge \text{Agent}(e, \text{John}') \wedge \text{Patient}(e, \text{b})$

This says that there is some event e of kicking, whose agent and patient were John and the ball b, respectively. Manner adverbs and many other adjuncts can also be expressed as predicates over the eventuality variable.

Parsons' presentation of event semantics is informal. The following formalization is based on Landman's (2000) event semantics, incorporating its essential ingredients but with certain simplifications where this is sufficient for our purposes.[11] We assume an extensional semantics throughout.

(9) **Outline of a model theory for events**

a. Our model contains two (disjoint) types of objects: The set of ordinary individuals D (including plural individuals, with the usual lattice structure), and a set E whose elements denote eventualities. The type of their elements will be denoted by e and s, respectively.[12]

b. There is a set V of unary *event type* predicates over the elements of E.

c. There is a finite set R of *roles* (*Agent, Patient,* etc.), which are relations between the sets E and D. Roles are predicates of type <e, st>. A thematic role r is *instantiated* for some event e if $r(e, x)$ holds for some individual x.

d. Unique role requirement: If a thematic role is instantiated for a particular event, then it is instantiated by exactly one (possibly plural) individual.

e. Role specification: Lexical constraints (general and/or lexeme-specific) determine which roles occur, either optionally or obligatorily, with each verb.

f. The eventuality variable is eventually bound by existential closure. Landman leaves the details

[11] I chose Landman's system because it sets down the mechanism of syntactic composition in some detail, and because it keeps theta role predicates with the verb instead of distributing them to its syntactic arguments.

While Landman's goal was to deal with the properties of plurals, these are not relevant for our purposes and are not discussed here. The reader is referred to Landman (2000:41–55) for details.

I deviate in several ways from Landman's notation. In particular, various objects and types are given different names here, the term *event type* is used differently (as a unary predicate only), sets of events are written in lambda notation (i.e., as characteristic functions) rather than with set brackets, and theta roles are written as relations between eventualities and individuals (after Parsons), not as partial functions.

[12] Landman defines the domains for types e and s as $D \cup \{\perp\}$ and $E \cup \{\perp\}$, respectively, allowing expressions of type e and s to be undefined.

open, allowing existential closure to apply at the VP or the IP level, depending on one's syntactic analysis.

The next question is how event types and roles enter the derivation. In line with the general approach of the Theta System and general considerations of compositionality, our account of lexical meaning is local: The semantic contribution of the verb must be inserted at only one place in the derivation, that is, thematic role information and the core denotation of the verb should form a single constituent. In particular, theta roles are associated with V^0, not with the argument NP that satisfies them.

While Landman's (2000) event semantics is purely local in the present sense, this is not the only way to approach the semantics of thematic relationships. Krifka (1989) does not include theta role information in the semantics of the verb; instead, it is assigned to the argument NPs and associated with the verb by means of open variables. The verb does carry syntactic information that controls the projection of its arguments. In later work, Krifka (1992) uses theta roles as selectional features on the verb, which must match corresponding features on its arguments; once again the corresponding semantics are only associated with the verb. Krifka adopted this architecture in order to allow flexibility in the realization of arguments and adjuncts. The present proposal accomplishes this goal while maintaining locality of lexical meaning.

Examples of an event type and a theta role predicate are given in (10a) and (b), respectively. Verb denotations are conjunctions of an event type and one or more theta role predicates in the appropriate order, as in examples (11a-b).

(10) a. $\lambda e\, \text{kick}(e)$.

b. $\lambda x_e\, \lambda e_s\, \text{Agent}(e, x)$.

(11) a. *kick* $\rightarrow \lambda y\, \lambda x\, \lambda e\, (\text{kick}(e) \wedge \text{Agent}(e, x) \wedge \text{Patient}(e, y))$

b. *run* $\rightarrow \lambda x\, \lambda e\, (\text{run}(e) \wedge \text{Agent}(e, x))$

This brief description should give the reader an idea of the formal properties we expect from our event semantics. While I use Landman's semantics for events as a starting point, there exist numerous other formalizations along similar lines (e.g., Link 1998:ch.10,11). The proposal developed below should be compatible with any of them.

3.1 Theta clusters as event modifiers

To embed the feature clusters of the Theta System in event semantics, we can simply use feature clusters as theta role predicates, i.e., as predicates over eventualities and ordinary individuals; we can write [+m+c](e, J) instead of *Agent*(e, J), etc., yielding formula (12) in place of (8b).

(12) $\exists e\,\text{kick}(e) \wedge$ [+c+m]$(e, \text{John}') \wedge$ [–c–m](e, b)

We can even continue to use *Agent* as an informal synonym for [+c+m], and write (8b) but read it as (12). This raises the question of the relationship between theta clusters and the traditional theta roles: Are they one and the same thing, or theta clusters just properties (features) of some richer object, the theta roles? For example, consider the roles Goal and Benefactor, both of which correspond to the [−c] cluster. Do we just have two names for the same thing, or do we have two distinct theta roles characterized by the same values for the c and m features? Note that the question is not specifically about event semantics: It arises as we relate the Theta System to any general account of theta roles.

Although Reinhart does not explicitly address the issue, she does stress that theta clusters have semantic content, and in her writings the theta clusters *are* the theta roles. Nevertheless her system is designed to be an adequate account of argument structure alternations, not of regularities in meaning, so it is possible that the nine cluster types she postulates are too broad for a complete semantic account of thematic roles, making a two-layer architecture necessary: Traditional theta roles are the real semantic objects, but in the computational system only the nine feature clusters of the Theta System need be distinguished. Thus, several distinct theta roles can correspond to the same feature cluster. This sort of two-tiered arrangement is suggested as a possibility by Chomsky (1981:139), who remarks that the theta criterion, as formulated there, need only apply to the "basic system of grammatical relations." But "If θ-roles are assigned in some different way outside this system, the principles we are considering can easily be modified to accommodate such cases while continuing to hold them as formulated here within the central subdomain that is our primary concern." In other words, the rules that govern the basic system of grammatical relations may be a simplified version of the full system. A similar state of affairs may well hold with respect to theta roles. Note that the question is *not* whether additional features should be added to the system, but whether the feature system appropriate for argument projection is less fine-grained than some other conception of theta roles that is better suited to doing semantics. The question is difficult to decide, not least because it is not at all clear which entailments belong in a

system of general-purpose theta roles and which should follow from the meaning of the verb itself.

If necessary, it would be technically straightforward to endow the set of roles R with a full inventory of semantic theta roles, which are mapped to the feature clusters of the Theta System by a projection function. I do not follow this option here since it would complicate the formalization of the theta system's operations. For simplicity and ease of exposition, I will assume that the theta clusters *are* the theta roles.[13] In the context of our model, *Agent* and [+c+m] are equivalent ways of writing the same theta role (element of R).

3.2 Events and concepts

If lexically derived verbal entries are produced via the application of general rules, they should be expressible in terms of the base entry. The question we must address here is: when we derive a new verb form in the lexicon, does it also require a new *event type*? For example, consider an event type predicate *pinch(e),* which is true of events of (transitive) pinching. We can use it to represent the meaning of the verb *pinch.* Now consider an event of self-pinching, e_0. Should the same event type predicate hold of it, or do we need a new predicate, *self-pinch(e),* to characterize such events? From the point of view of our event semantics, the simplest solution is to use the same event type: Events of self-pinching are events of pinching, distinguished by the fact that their Agent and Patient are the same individual. If we now imagine applying a lexical reflexivization operation on the verb *pinch,* the meaning of the resulting reflexive verb *self-pinch* (or *Refl(pinch)*) can be expressed by the same event type predicate, *pinch(e),* as follows:

(13) a. *self-pinch* = $\lambda x\, \lambda e$ $\text{pinch}(e) \wedge \text{Agent}(e, x) \wedge \text{Patient}(e, x)$

I will adopt this strategy for our model: Lexicon operations do not change the underlying event type predicate. Derived lexical entries can be viewed as compound expressions involving the base concept. In the simplest case, arity operations have no effect on the event type. But lexical derivations sometimes involve systematic meaning shifts that go beyond changing the inventory of theta roles.

[13] This is almost certainly insufficient if we adopt Marelj's (2002) system of disambiguating underspecified feature clusters into just four fully specified clusters (so that [+m] must be resolved into either [+c+m] or [−c+m]). The disambiguation process yields just four distinct types of theta cluster after lexical insertion, almost certainly too few to capture the diversity of theta role semantics. I consider it an open question whether the full system of nine clusters is adequate as a complete system of semantic roles.

Middle constructions, for example, can involve aspectual changes. To the extent that arity operations, or other operations on lexical items, have predictable effects on the core meaning, it should be possible to express the meaning of the derived forms as some complex expression involving the basic event type predicate. But such effects are beyond the scope of most of Reinhart's work on the Theta System, which focuses on the effects of arity operations on arguments. (In fact Reinhart shows that aspectual factors are *not* involved in accounting for the causative-unaccusative alternation.) I have nothing to add to the topic, and will therefore treat arity operations as if they had trivial effect on the event type predicate.

However, lexically derived verbs frequently have idiosyncratic meanings; for example, the Greek verb *tsakono* 'to catch' has a special meaning for its reciprocal form, *tsakonome* 'to quarrel'. We must therefore distinguish two cases: Derived forms with unpredictable meanings are listed in the mental lexicon; their meaning can in principle involve any event type, regardless of the meaning of the base verb. Derived forms with compositional meaning are not listed; the definition of the lexical operation must specify a way of computing the meaning of the derived verb as a function of the meaning of the base verb.

I assume here a notion of "listedness" as proposed by Di Sciullo and Williams (1987). Not only roots but entities of arbitrary complexity, including phrases (idioms), may be listed in the lexicon. The listed objects are those whose meaning is not compositionally predictable, and a listed meaning overrides ("blocks") the would-be compositional meaning of the listed entity. A reciprocal verb with idiosyncratic meaning, then, is a relatively simple example of a non-root form that is listed in the mental lexicon.

3.3 The order of projected arguments

Having discussed the interpretation of theta clusters in an event semantics model, we now consider the complete verb. Our goal is to derive verb denotation functions like (11c-d) above, while allowing the machinery of the Theta System to control the "role specification" functions that Landman assumes. It is not enough to simply include complete forms like (11c) or (d) in the stored lexical entry of each verb: First, alternating verbs need a different function for each mode of argument realization, and listing all of them in the lexicon would lead to undesirable redundancy; second, each such formula specifies the order in which the arguments of the verb in question are projected. This is in conflict with the lexical model of the Theta System, which derives alternations in the argument inventory of verbs from a single

underlying lexical entry, and computes their order of projection on the basis of general principles. Our goal, then, is to come up with a suitable model of lexical entries in the mental lexicon, which the mechanisms of the Theta System can manipulate to produce the various desired denotation functions.

We begin with the question of whether the verb's lexical entry includes information on the ordering of its thematic grid. As we have seen, the Theta System includes a general procedure, the CS merging instructions given in (3), that (partially) determines the syntactic projection of any verb's arguments. This accounts for the well-known fact that thematic roles are not mapped idiosyncratically to verbal argument positions; their arrangement is subject to universal constraints.

If the order of arguments was stored in the lexical entry of the verb, there would be no need to calculate it. Still, global constraints on ordering are not incompatible with having lexical entries that specify the argument order of each verb: the general constraints could then take the form of well-formedness constraints on lexical entries, i.e., of a filter. In other words, lexical entries might hardwire the order of their arguments, but only those orders consistent with the CS-merging instructions would be permitted. Such a "declarative" account, however, would be at odds with the Theta System's strongly procedural orientation. The CS merging instructions apply after lexicon marking and lexical arity operations, not directly to the stored lexical entries. They could not be easily recast as well-formedness constraints on the stored representations. To leave room for the Theta System's processes to apply, we must start with a lexical entry whose arguments have no intrinsic order; the CS merging instructions eventually build an ordered representation, perhaps after the application of lexical operations that involve the addition, deletion or modification of some clusters of the original lexical entry.

We arrive at the same conclusion from considerations of redundancy: Following the common view of the mental dictionary as the repository of *unpredictable* information about lexemes (e.g., see Di Sciullo and Williams 1987:3,14), it may be argued that since the order of arguments is predictable, it does not need to be stored in the mental dictionary. Besides, a system that lists order in the lexical entry could allow exceptions to exist, such as verbs whose Agent and Patient occur in reverse order. But such verbs are unattested. We conclude:

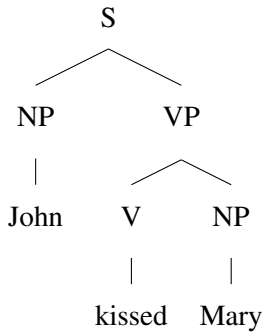
(14) The order of thematic arguments of a verb is not specified in its stored lexical entry.

Recall that our objective is to provide an explicit account of the semantics of verbs as they

undergo arity operations, from the stored lexical entry to the final interpretable form. The end result of this process will be a model-theoretic denotation, in the form of a truth-valued function over individuals and events. But in the standard formalisms of model-theoretic semantics, Montague grammar, etc., function arguments are inherently ordered: The denotation of *kick* (example (11d)) specifies that the first (internal) argument is its Patient, and the higher argument is its Agent. This is more specific than the lexical entry should encode, since we have decided to leave our lexical entries unspecified for argument order.

This predicament is not an accident of our choice of formal framework: fixed argument order is inherent in syntactic and semantic phenomena, and hence also in the theoretical frameworks employed in their analysis. In syntax, constituents combine in a particular order that is expressed in a hierarchical (structural) way. In a given syntactic tree, each constituent appears in just one structural position, which determines the order in which it forms a constituent with other parts of the tree. (A tree, of course, can be transformed into a different tree through movement.) In particular, clause structure determines the order in which verbal arguments combine with the verb and other arguments. The functions of our semantic framework similarly fix the order in which the arguments of the verb will be saturated. For example, the tree in (15) specifies the syntactic projection of the arguments of the verb *kissed.* Its constituent structure determines the order in which the semantic counterparts of its constituents will combine. Arguments are assigned to the appropriate thematic role because the semantic form at each tree node indicates the order in which its arguments should be saturated; for example, the semantic form of the verb *kiss* specifies that the first argument it combines with is the Theme rather than the Agent.

(15)



For our model of the Theta System, this means that listing a particular denotation function in the lexicon would undesirably encode a certain order of argument projection in the lexical entry. Granted, it is not a difficult technical task to transmute one order of argument projection into another. But listing

one particular order in the lexicon, even an arbitrary one, goes against the principles of the Theta System— especially since there is no evidence for arity operations that rearrange the order of argument projection.

3.4 Manipulating theta roles

The considerations of the previous section lead us to adopt a semantic object more articulated than simple denotation functions: The inventory of theta roles must be kept in the form of a set, with no explicit order. This also turns out to be useful for another reason: *Some* of the Theta System's arity operations perform manipulations on theta roles that are impossible to carry out, in a general way, on the arguments of model-theoretic functions. Two operations fall in this category: *Decausativization* (called "expletivization" in Reinhart 2002) and *causativization.*

Decausativization completely eliminates a [+c] (Cause) theta role, and is responsible for creating unaccusatives under the causative-unaccusative alternation (see section 2.2). It also removes a verb's Accusative case feature, as indicated by its disappearance in the schema below.

(16) Decausativization: Reduction of a [+c] role

$\mathrm{V}_{+\mathrm{Acc}}$ ([+c], θ_j) $\rightarrow$ V (θ_j)

To carry out decausativization, our system must not only delete a theta cluster but also recognize the presence of a particular cluster, [+c], sufficiently well to ensure that decausativization only applies to verbs that have it.

It should be stressed that complete elimination of a theta role ("reduction," in the Theta System's terminology) is truth-conditionally distinct from existential closure ("saturation") of the same argument position, which as we will see is easy to define as a compositional operation on functions. If we apply decausativization (i.e., reduction) to the transitive from of *break*, the result is as shown in (17b). As we have seen, it represents the unaccusative variant of *break.* If we apply saturation we get (17c), which represents the passive form *was broken*:

(17) a. *break* = $\lambda x\, \lambda y\, \lambda e\, \mathrm{break}(e) \wedge \mathrm{Cause}(e, y) \wedge \mathrm{Patient}(e, x)$

b. *break*$_{\mathrm{Red}}$ = $\lambda x\, \lambda e\, \mathrm{break}(e) \wedge \mathrm{Patient}(e, x)$ (Reduction)

c. *break*$_{\mathrm{Sat}}$ = $\lambda x\, \lambda e\, \exists y\, \mathrm{break}(e) \wedge \mathrm{Cause}(e, y) \wedge \mathrm{Patient}(e, x)$ (Saturation)

Both (17b) and (17c) are of type <e, st>: when combined with an individual (type e), they each give us

a predicate over events. The difference is that unaccusative (17b) can be satisfied by any event of breaking, whether or not it has a causal participant; but (17c) can only be satisfied by events of breaking that do have a causal participant. This accounts for the well-known fact that unaccusatives and passives contrast syntactically, with only the latter allowing purpose clauses, *by-phrases*, etc.[14]

Causativization, which despite the name is not the opposite of decausativization, adds a [+c+m] role and modifies any existing cluster carrying the +c feature, replacing it with its corresponding −c cluster. It is responsible for constructions like *walk the dog.*

(18) Causativization (Reinhart 2002)

a. Feature adjustment:[15] Change a +c feature to a −c feature.

E.g., walk([+c+m]) → walk([−c+m])

b. Agentivize: Add an agent role.

E.g., walk([−c+m]) → walk([+c+m],[−c+m])

The two arity operations require access to the inventory of theta roles in several different ways. Decausativization can only apply to verb entries that have a [+c] cluster, so the machinery of the theta system must be sensitive to its presence: Either it should be possible to directly detect the presence of the [+c] cluster, or decausativization must be so defined that it fails if applied to a verb lacking it. While causativization is in principle compatible with any theta grid, it too must be able to identify the presence of particular theta roles sufficiently well to target them for feature adjustment. Finally, both arity operations must be able to remove or replace the targeted roles. We will now see that if we model our verb denotations as model-theoretic functions in an ordinary type logic, there is no general way to carry out these operations. The reason is that functions, including verb denotations, are not hierarchically structured objects or strings of symbols. Formally, functions are defined simply as mappings from one set to another; in our case, from one set of functions to a set of functions of another

[14] The contrast is often illustrated with the following minimal pair (Manzini 1983):

(i) * The boat sank PRO to collect the insurance.

(ii) The boat was sunk PRO to collect the insurance.

The matter is in fact more complex than we can discuss here, as a reviewer points out; a review of the empirical motivations for the Theta System's arity operations lies beyond our present goals.

[15] Reinhart (2002) actually refers to this step as "decausativize". The name was changed to "feature adjustment" somewhere around 2006, in order to avoid confusion with the arity operation decausativization. The updated formulation, which includes some other changes, can be found in Rákosi (2006:38).

type-logical type. As such, a function is harder to manipulate, as a formal object, than a collection of distinct theta clusters and other features.

Mathematically, a function is a set of ordered pairs, with each element in the domain of the function being paired with the function's value for that element.[16] At this level there is no reflection of how the value is calculated from the argument, or even a notion of calculation of any sort. The truth-valued function λe run$(e) \wedge$ Agent(e, x) is simply a "characteristic function" that returns *True* for all events of running whose Agent is x (whoever that is). Listing the set of these events would convey exactly the same information. The denotation of *run,* $\lambda x\, \lambda e$ run$(e) \wedge$ Agent(e, x), is simply a function that maps any individual x to the corresponding characteristic function. This function could be written in other ways (for example, as a very large truth table). The characteristic functions it gives us (after combining with particular individuals) might be defined in other ways, without mentioning *Agent* in their definition. Suppose that whenever John runs, he sweats; that events of running are the only occasions on which he sweats; and that our model represents such co-occurring events as a single eventuality. *Sweat* is an emission verb, with a single Patient argument (Reinhart 2002). The following expressions would then pick out the same set of events, i.e., they *are* the same function in this model:

(19) a. λe run$(e) \wedge$ Agent(e, j)

b. λe sweat$(e) \wedge$ Patient(e, j)

Consequently it is not possible to have an operator guaranteed to recognize all and only the functions containing *Agent* as one of the conjuncts in their definition: If applied to function (19), our hypothetical operator would have the impossible task of choosing between forms (19a) and (19b). Since the same function can be defined either with or without using Agent, the task (if it is meaningful at all) assumes knowledge of how a function was in fact defined; and this is not present in the function itself.

The same is true of complete verb denotations: If in our model the events of running are exactly the events of sweating (and, reasonably, the person running is the one sweating), the following expressions describe the same function of type <e, st>:

(20) a. $\lambda x\, \lambda e$ run$(e) \wedge$ Agent(e, x)

b. $\lambda x\, \lambda e$ sweat$(e) \wedge$ Patient(e, x)

Any equivalent formulation would be a valid way of expressing the same function. The presence

[16] By definition, a function must supply exactly one value for each element in its domain.

or absence of an *Agent* argument can only be detected through the truth-conditional properties of a verb's denotation, and this turns out to be impossible: Such denotations simply do not contain enough information. Denotation functions can be written as lambda forms, as above, but the symbols constituting the lambda form are not actually part of the function.

It might seem that this problem can be avoided by switching to an intensional (possible-worlds) model. In the Appendix I show that this is not the case: Intensional models are also liable to having the same model-theoretic function arise by more than one route.

We could sidestep the problem, of course, by adopting a richer view of functions. For example, we could define functions as strings of symbols comprising a lambda form like the ones in (20). It would then be possible to define operations that cut and splice arbitrary parts of the function. But such a move would have far-reaching consequences, and is in any event much more drastic than our present proposal of an expanded structure for the limited purpose of lexical derivation. The point is that functions *as standardly defined* are restricted in the ways we have detailed.

Detecting the presence of a theta role is only the first hurdle. Even if we could assume that a particular theta role is present in a verb's denotation, there is no systematic way to delete it. For example, we cannot map all agentive verb denotations to equivalent denotations that do not involve an Agent. The reason is essentially the same as for the problem of detecting the presence of a role: Since a function does not explicitly contain the rules that created it, it is not possible to reconstruct, and omit, the truth-conditional contribution of the Agent predicate.

Proving that removal of a theta role is impossible involves a fair amount of technical detail, so I defer it to the Appendix. Briefly, the reason is that there are simply too many possible answers. We can informally illustrate the problem with an analogy: Consider a function d that gives us the straight-line distance of any geographical location from some fixed point (say, a particular location in Athens). Suppose now that I give you a distance, 42,195 kilometers, and I ask you to figure out the point whose distance I calculated to arrive at this number. The problem does not have an answer because there are a great many points (forming a circle around our reference point) for which this number is the answer. The task of going from an agentive denotation function to its Agent-less counterpart is impossible for the same reason: There are too many possible answers. If we think of the verb denotation as being assembled by adding theta roles to the event type one by one, the operation of adding a theta role can be viewed as a function that maps our partially built functions to other functions with one more

argument; <e, st> to <e, <e, st>>, in the case of adding a second theta role. But this function is many-to-one, and therefore cannot be inverted. It is hoped that the correctness of these claims will be apparent to many readers. I provide more discussion along with the proofs in the Appendix.

The following are natural consequences of the standard treatment of denotation functions:

(21) **Limitations of operations on verb denotations**

a. A theta role cannot be completely deleted from a verb denotation, nor replaced with another one.

b. No operation on denotations can be restricted to particular thematic types.

(22) **Operations that can be readily applied** to the arguments of model-theoretic functions

a. Existential closure

b. Reordering

c. Identification with another argument (as in reflexivization).

d. Insertion of additional arguments

The limitations in (21) are dicsussed above. To confirm that the operations in (22) are easy to define as operations on functions, we provide formulas that will perform them on the denotation of transitive verbs:

(23) a. SAT(V) = $\lambda V\,.\,\lambda x\,\lambda e\,\exists y\,V(x)(y)(e)$

b. FLIP(V) = $\lambda V\,.\,\lambda y\,\lambda x\,\lambda e\,V(x)(y)(e)$

c. REFL(V) = $\lambda V\,.\,\lambda x\,\lambda e\,V(x)(x)(e)$

d. EXP(V, R) = $\lambda V\,\lambda R\,.\,\lambda x\,\lambda z\,\lambda y\,\lambda e\,[V(x)(y)(e) \wedge R(z)(e)]$

In the above, V is a transitive verb denotation (type <e, <e, st>>) and R is a theta role predicate (type <e, st>). Each of the four formulas carries out the corresponding operation described in (22). *SAT* performs existential closure on the external argument, *FLIP* reverses the order of projection of the two arguments, *REFL* causes both theta roles to be satisfied by a single projected argument, and *EXP* (for *expansion*) adds a third theta role, which is projected just below the highest argument. Analogous formulas can be defined for other argument positions and for intransitive verbs, ditransitives, or functions of any other type.

4 A model for word derivation from lexical entries

We have seen that the Theta System's model of lexical derivation requires free access to the inventory of theta roles, and assumes that argument order is initially indeterminate. Neither of these is a property of the usual type-logical functions of model theory. But during syntactic derivation and semantic interpretation, we are dealing with objects in a form that does impose an explicit hierarchy (order) on the verb's arguments. It follows that the verb's stored lexical entry is *not* in that form. We are led to the conclusion that there is a *change of representation,* which I refer to as *assembly,* somewhere between the lexical entry and the LF interface. To make the proposal maximally conservative, we should try to locate assembly as early as possible. I will show that this point is at the transition between the lexical and the syntactic component.

In this section, I provide an explicit model that is capable of undergoing the operations specified by the Theta System, and a procedure for assembling it into a single denotation function of event semantics. In section 4.6 we look in more detail at when assembly takes place, and its consequences.

We have arrived at the following scenario: In the stored lexical entry, the theta grid of the verb is an unordered set of feature clusters. This set is manipulated by arity operations, which may add, remove, or replace the clusters in the set. At a certain point the order and manner of argument projection (internal or external) is determined, and the derived lexical object is converted into a different form, expressible as a single denotation function, which gives the proper event semantics and, necessarily, imposes a fixed argument order. Along with the inventory of theta roles, each stage in the process must encode all sorts of other information that must be kept track of at that level: the lexical entry might include a phonological representation, morphological properties such as conjugation class and compatibility with various derivational morphemes, etc., while a syntactic representation must, among other things, include interpretable and uninterpretable features as required by the Minimalist architecture. But each form must also include information about the verb's meaning and about its arguments. It is this information we focus on, with the understanding that it is just part of a larger system.

4.1 The lexical entry

We may model the stored lexical entry of the verb as including the following (along with other information which will not concern us).

(24) a. A one-place predicate over eventualities (type <s, t>), representing the core verb meaning.[17]

b. A set T, whose elements represent the theta roles of the verb and are drawn from the set of theta clusters.

For example, the relevant part of the lexical entry for *kiss* is as follows:

(25) **Lexical entry for *kiss***

Event type: $\lambda e\,\text{kiss}(e)$

Roles: { [+c+m], [−c−m] }

In combination with the general constraint that Agents must project higher than Themes, our lexical entry contains enough information to construct the denotation function of the verb *kiss*:

(26) $\lambda x\,\lambda y\,\lambda e\,\text{kiss}(e) \wedge \text{Agent}(e, y) \wedge \text{Theme}(e, x)$

During the course of lexical derivation, an object representing a copy of the stored lexical entry is modified in various ways. We will refer to this object, initially a copy of the stored lexical entry, as our *working structure, WS.*[18] In addition to the information inherited from the lexical entry, the WS can store values for the feature ACC (assign accusative case), which is introduced by lexicon marking rule

[17] For simplicity, I will assume that the core verb meanings correspond to event types, i.e., are elements of the set V defined in (9). It is possible, however, for the relationship to be non-trivial. If to smash means to break with great force, the lexical entry for *smash* might define it as a combination of the event types *break* and *forceful* (a manner predicate), eliminating the need for an event type *smash* in our semantics:

(i) *smash* = $\lambda x\,\lambda y\,\text{break}(e) \wedge \text{forceful}(e) \wedge \text{Patient}(e, x) \wedge \text{Cause}(e, y)$

We can use a similar approach for any verbs whose meaning is based on some event type but with aspectual or other modifications. For example the event type predicate *open,* with appropriate additional conditions, might express the meaning of the verb *reopen.* But note that our simple event semantics does not encode time or the aspectual dimension of events, hence there is no way to provide an explicit account without an extension of our semantic model. (For a more considered treatment of the prefix *re-*, see Williams, this volume).

[18] The WS differs from the stored lexical entry in that it can carry information created or computed in the course of the derivation. It is the lexicon equivalent of the (initially incomplete) syntactic structures that are manipulated by Merge during syntactic derivation, referred to as *structural descriptions* (SD) by Chomsky (1995:14ff) or simply as *syntactic objects* by Adger (2000) and Hornstein, Nunes, and Grohmann (2005). The WS, in short, is a sub-lexical SD.

(2c). Arity operations can add, suppress or modify elements in our working copy of T,, and may delete the *Acc* feature again. Therefore the WS can be modeled as a tuple $<P, T, \Phi>$, where P is an event type predicate, T is a set of theta roles, and Φ is a set of features.

What happens when we assemble (26) from the WS? After assembly, theta structure information might be present twice, in the original theta grid and in the generated denotation function. To ensure non-redundancy and consistency, I will assume that the same information is not represented in two places at the same time: the ordered form replaces the original, unordered information. The relevant content of the lexical entry is consumed ("checked") during assembly, and is no longer accessible directly. Any subsequent morphosyntactic or semantic operations must work by manipulating the new representation. In particular, any arity operations that apply after assembly can modify argument structure only by manipulating the assembled denotation function. Accordingly, their semantic effects should be expressible as operations on functions. To summarize,

(27) **Non-redundancy principle:** The set of theta roles is assembled into a unified denotation by "checking" each role, which consequently loses its independent existence and is no longer accessible except as part of the resulting denotation function.

Although a model-theoretic function incorporates ordering information, it does not contain any syntactic information about the projection of its arguments beyond their order: The semantic denotations for an unaccusative and an unergative verb are combinatorially indistinguishable (type <e, st>), and two-argument unaccusatives are similarly indistinguishable from transitive verbs with one internal and one external argument. I will assume that when the set of theta roles is assembled into a denotation function, the necessary projection information is encoded in the form of syntactic features that are stored with the assembled lexical entry.

Word derivation in the Theta System proceeds in a sequential manner, with constraints on the sequence of operations doing much of the work of obtaining the right result. Saturation and decausativization (reduction) apply after introduction of Case features and internal–external indices ("lexicon marking"). In addition to their effect on theta clusters, saturation and reduction can have effects on argument projection and licensing, which rely on the precise sequence of events. As detailed in section 2.2, non-derived intransitives do not undergo marking and are therefore always unergative; but a verb that becomes intransitive because of reduction (or saturation) already carries projection

indices, and could therefore be unaccusative. Causativization, on the other hand, applies before marking and therefore affects ("feeds") marking and the introduction of Case features.

Our model for lexical derivation faithfully follows the prescriptions of Reinhart (2000, 2002), Reinhart and Siloni (2005). The general outline proceeds as follows. Its components are described in more detail below.

(28) **The derivation process**

1. Copy the verb's stored lexical entry into a *working structure,* WS.
2. Apply arity operations, the calculation of the ACC feature, and the lexical marking procedure as specified by Reinhart (2002).
3. Immediately before the assembly step, determine whether an external argument will be projected.
4. Perform the assembly step, expressing each cluster as a theta role predicate and conjoining them with the event type in the appropriate order.
5. All further operations must apply to the assembled denotation function, plus any remaining information such as the ACC feature.

4.2 Arity operations

The following arity operations constitute the inventory of universal theta role manipulations, employed by constructions that manipulate argument structure. Because of the change of representation involved in the assembly step, pre- and post- assembly arity operators will manipulate different objects, and must be defined differently.

(29) a. **Saturation** is the operation that applies (inter alia) in passive formation. It existentially closes a theta role, which is consequently present for semantic interpretation, but is normally not realized syntactically. (It can be reintroduced as an oblique, or license instrument adjuncts or other constructions that require it.)

b. **Arbitrarization** is a variant of saturation that involves existential closure by a variable marked for "arbitrary human" reference (Chierchia 1995). It is involved in the derivation of middle verbs. Arbitrarization is not in the original inventory of arity operations for the Theta

System; see Marelj (2004), Reinhart and Siloni (2005).

c. **Decausativization** involves reduction (complete elimination) of the external theta role. It can only apply in the lexicon. (By definition, based on its observed distribution.) It must apply to a multi-argument verb entry that has a [+c] theta cluster as one of its arguments; that argument is removed. Decausativization is responsible, inter alia, for deriving unaccusatives such as *break, open* from their transitive counterparts.

d. **Reflexivization** identifies two roles, creating a verb in which one argument fulfils two thematic roles. It can apply either in the lexicon or in the syntax.[19]

e. **Causativization** adds an Agent [+c+m] role. If the original lexical entry projected a cluster with a +c feature (an agent, cause or instrument), this is changed into the corresponding −c cluster. Like decausativization, causativization can only apply in the lexicon.[20]

We now define these arity operations as manipulations of the WS. Since (as I will show) syntactic arity operations do not have access to the WS, we only provide the definitions appropriate for the lexical component.[21]

(30) **Lexical arity operations on the WS.**

a. **Saturation**

(i) Replace a cluster α with the result of applying existential closure on its individual argument; its meaning is defined as $\lambda e\, \exists x\, \alpha(x)(e)$.

(ii) Remove the ACC feature if it is present.

b. **Arbitrarization**

(i) Replace an element α with the result of applying existential closure with a variable designated for arbitrary human reference, as defined by Chierchia (1995); its meaning is

[19] This description reflects the analysis in (Reinhart and Siloni 2005) and later. Reinhart (2002) analyzes reflexivization as internal reduction, the elimination of the internal theta role.

[20] The Theta System limits its treatment of causation to the combinatorial properties of the +c clusters; in particular, it does not articulate a relationship between a causing and a caused event, or equivalent. I assume that an analysis along these lines can be made compatible with the Theta System, but in fact this might not be necessary: Neeleman and van de Koot (this volume) argue that while causation is logically a relationship between events, a Cause argument is still the right way to represent the *lexical semantics* of causation (see their paper for the precise statement of what this claim entails).

[21] For those operations that can apply in either component (saturation, reflexivization), the main practical difference is that the syntax versions do not suppress the Accusative feature.

defined as $\lambda e\, \exists x_{arb}\, \alpha(x)(e)$.

(ii) Remove the ACC feature from Φ, if it is present.

c. **Decausativization**

This operation can only apply if T contains the cluster [+c].

(i) Remove the cluster [+c] from T.

(ii) Remove the ACC feature, if it is present.

d. **Reflexivization**

(i) Replace two clusters α, β with the bundled cluster $\alpha\beta$, whose meaning is given by the formula $\lambda x\, \lambda e\, \alpha(x)(e) \wedge \beta(x)(e)$.

(ii) Remove the ACC feature from Φ, if it is present.

e. **Causativization**

(i) If there is a cluster containing the +c feature, replace it with the corresponding −c cluster: [+c+m] becomes [−c+m], etc.

(ii) Add a [+c+m] cluster to T.

(iii) Add an ACC feature, if appropriate.[22]

After the application of the arity operations, the possible elements of T in the WS are the following. (We assume that only elementary clusters occur in stored lexical entries.)

(31) a. The nine clusters of the theta system.

b. For each cluster α, counterparts *Sat(α)* and *Arb(α)* generated by saturation and arbitrarization, with semantic type <s, t> and meaning defined as above.

c. For each (unordered) pair of clusters α and β, a "bundled" element $\alpha\beta$ with semantic type <e, st> and meaning defined as above.

For example, the bundled role <[+c+m][−c−m]> is produced by reflexivization of a verb with the theta clusters [+c+m] and [−c−m]. Rule (d) above gives its interpretation as $\lambda x\, \lambda e\, [+\text{c}+\text{m}](x)(e) \wedge [-\text{c}-\text{m}](x)(e)$, i.e., $\lambda x\, \lambda e\, \text{Agent}(x)(e) \wedge \text{Patient}(x)(e)$.

[22]The conditions for adding the ACC feature are spelled out by the Theta System's lexicon marking rules (2).

4.3 Internal and external projection

Recall that the semantic framework does not distinguish between internal and external arguments. A function incorporates ordering information, but does not determine the syntactic position of its arguments: unaccusatives and unergatives are both just predicates of type <e, st>, and two-argument unaccusatives are similarly indistinguishable from ordinary transitives. In the framework of the Theta System, the CS merging instructions determine the type of projection, based on lexicon marking of the theta clusters and other considerations (see (2) and (3)). I will assume that whether or not to project an external argument is a syntactic property. The lexical component is responsible only for deciding whether the verb should do so, and for passing on this information to the syntax in the form of a feature.[23] This feature, which I will call EXT, is not part of the Theta System as formulated by Reinhart and Siloni; I introduce it here as a technical means of preserving the decision computed by the CS marking rules, since the information on which they depend becomes inaccessible once the external theta role is assembled into the denotation of the verb. Like ACC, EXT is a simple privative feature: it is either present or absent.

While the theory of the Theta System includes a detailed account of when external projection will take place, the relevant rules are subject to elaborate rule ordering constraints as discussed earlier: The lexicon marking rules (2), which will not apply to verbs with only one argument, do apply to verbs that have gained a second argument through the causativization operation. Therefore causativization must precede the application of the marking rules. But the marking rules are not blocked by reduction, saturation and reflexivization, which suppress an argument position; a verb left with just one argument due to their application will retain its marking index, if any.

[23] Chierchia (1989/2004) shows that it is possible to encode the difference between internal and external projection in the semantics, by means of an alternative semantic formalization. Chierchia's system gives predicates an atomic type, $<\pi>$. A "predication" operator, "$^{\cup}$", converts predicates to functions of type <e,p>, where <p> is the type of propositions. Because the predication operation is assigned to the I^0 projection, its output must combine with an argument external to IP (hence, an external argument). The distinction between $<\pi>$ and <e, p> can be exploited to control the projection of arguments.

Kratzer (1996) adopts another approach to this issue: Following Marantz's (1984) suggestion that the external argument is not part of a verb's theta grid at all, she develops a neo-Davidsonian event semantics for verbs that does not need to separately encode the internal–external distinction in the syntax. This solution is incompatible with the Theta System, which crucially relies on including the external argument in the verb's theta grid.

A fully faithful reconstruction of the Theta System would include a way to represent clusters marked with the indices 1 and 2. These indices, assigned by the lexicon marking rules and controlling external vs. internal projection, are not explicitly represented in the model developed here. The reason is that the indices are fully predictable from the features of each cluster, being basically shorthand for being an all-plus or an all-minus cluster (indices 1 and 2, respectively). Rather than storing the indices in the theta grid T and reading them off again, we will achieve the same effect by direct reference to the values of the theta clusters. At the point where the Theta System specifies that indexing takes place, it is sufficient to determine whether it *should* take place. As far as I can see, the change simplifies the workings of the system but has no effect on its empirical predictions. The choice is largely a matter of personal preference, however, and readers who find the orthodox version preferable can easily fall back on it.

The sequence of events, therefore, must be the following:

1. Apply causativization (if applicable).
2. Decide, based on the number of arguments, whether the marking rules will apply.
3. Apply other arity operations (if applicable).
4. Apply the marking rules (if allowed by step 2) and the CS merging instructions.

The simplest way to carry out this procedure is by means of another feature, MARK, added to the WS in step 2 if its theta grid contains more than one argument at that point. Step 4 will then apply the marking rules only if the feature MARK is present.[24]

There is certainly more to argument order than the decision of which argument, if any, should project externally. The theory of the Theta System only imposes weak constraints on argument

[24] The manuscript version of this text adopted an insight by Siloni (2002), who points out that we can account for the different impact of the arity operations if we stipulate that the marking rules apply to the "maximal" theta grid of the verb: causativization, which extends the theta grid, is visible when deciding whether to mark, while operations that reduce it are not. While this is in my opinion preferable on conceptual grounds, the criterion is not trivial to apply, since our sequential model can eliminate a cluster: at the point where the marking rules are applied, it is necessary to know whether a cluster was present but has been deleted. We can eliminate step 2, then, but not the feature MARK. Instead of adding it at a single point in the derivation, it would be added at any point at which the WS contains more than one argument.

Adopting this approach would also complicate our presentation, since all aspects of the revised system must be precisely defined. Here I follow the standard formulation of the required sequence of events, which allows us to simply assert that all operations (modulo details of the implementation) are ordered and carried out as specified by the Theta System.

ordering, focusing mostly on the question of internal vs. external projection. It is assumed that there are other factors, some unknown and some possibly idiosyncratic, that further constrain the order in which arguments are projected. Whether any true optionality remains after such factors are considered is unclear. Our system is equally consistent with either state of affairs. The CS merging instructions compute constraints on argument projection; if more than one order would be grammatical, an order is arbitrarily selected either by the marking and merging process or (implicitly) during assembly.

(32) CS merging instructions (partial description)

a. When nothing rules this out, merge an argument externally.

b. If the WS contains the feature MARK, an argument realizing an all-minus feature cluster must merge internally. An argument with an all-plus cluster must always merge externally. Bundled clusters must merge externally if and only if one of the bundled roles is an all-plus cluster.

c. In accordance with instruction (a), an argument will merge externally unless:

(i) some other argument is already merging externally,

(ii) instruction (b) requires it to merge internally, or

(iii) the verb has the ACC feature, and no other argument is available to receive Accusative case.

d. If some argument can merge externally, add the EXT feature to the WS.

We use the simple privative feature EXT to record the outcome of the CS merging algorithm: If it is present, project the highest argument externally, otherwise project internal arguments only. Nothing hinges on this detail, however: readers are welcome to substitute any other feature assumed to encode externality in their model of syntactic derivation.

The above formulation duplicates the effect of Reinhart's original lexicon marking rules and CS merging instructions. As discussed in section 2.2, her writings discuss some additional conditions that we will not give a full account of. Reinhart holds, inter alia, that certain co-occurrence conditions render the realization of some clusters optional or in some cases impossible. We therefore allow the following, without going into the details:

(33) **Non-realization**

Under the optional or obligatory conditions spelled out by Reinhart (2002, forthcoming), a cluster may be either deleted from T or existentially closed prior to assembly.

4.4 The assembly step

The assembly step involves assembling the verb's theta clusters, in the proper order, and the verb's event type predicate into a single logical formula. To accomplish this we recursively define a series of argument projection operators, $\textit{Proj}_n$, which add a theta role predicate to a partially constructed verb denotation.[25] The elementary one, $\textit{Proj}_0$, adds a theta role to an event type predicate. Higher order projection operators operate on denotations that already include some arguments.

(34) **Argument projection operators**

a. $\textit{Proj}_0 = \lambda R_{e,st}\, \lambda P_{st}\, [\lambda x_e\, \lambda e_s\, P(e) \wedge R(x, e)]$.

b. $\textit{Proj}_1 = \lambda R_{e,st}\, \lambda P_{e,st}\, [\lambda z_e\, \textit{Proj}_0(R, P(z))]$.

c. ...

d. $\textit{Proj}_n = \lambda R_{e,st}\, \lambda P_{e,...st}\, [\lambda z_e\, \textit{Proj}_{n-1}(R, P(z))]$.

The above are applied to a theta role predicate (the argument R) and a partially built verb denotation (the argument P). For the lexicon, we only need as many operators as the highest arity verb occuring in natural language. For ditransitive verbs, $\textit{Proj}_2$ is the highest needed.

We can then carry out assembly as follows:

(35) **Assembly procedure**

1. Begin with the verb's event type predicate.
2. If T contains any saturated clusters generated by (30a,b), combine them with the partial verb denotation through generalized conjunction.
3. Select the verb's theta clusters, one by one, in an order consistent with the CS merging instructions (lowest argument first). Use the appropriate projection operator to combine the theta cluster with the partial verb denotation, as follows:
   (a) For elementary clusters, merge the corresponding theta role predicate.
   (b) For bundled clusters, merge the bundled predicate defined by (30d).
4. Remove each cluster from T as it is used.

Generalized conjunction (step 2) combines two predicates of the same type, giving a predicate expressing a conjunction of their truth conditions over the same arguments. Because the result is also of

[25] The operators are modeled after the *Event Identification* rule of Kratzer (1996).

the same type, generalized conjunction can be used as many times as necessary. For predicates of type <s,t> (core verb denotations and saturated theta roles), it is expressed as follows:

(36) $P + Q \equiv \lambda e\,[P(e) \wedge Q(e)]$

The result of assembly is a function with a fixed hierarchical order for its arguments. It expresses the semantic content of the verb in the form of a traditional truth-conditional denotation. It enters the syntax embedded in a syntactic object of the sort assumed in Minimalist accounts (e.g., Adger 2000:17ff): a lexical head that also contains syntactic features controlling its behaviour during syntactic derivation. In particular, it should carry the features for accusative Case assignment and for whether to project an external argument (see section 4.3):

(37) If the features EXT and ACC are present in the WS, add them to the generated syntactic object.

The utilization of these features depends on one's model of syntactic derivation, and will not concern us here. Our model implements the lexical arity operations as manipulations on the WS, and then converts the WS into the usual kind of semantic and syntactic object. In the next section, we discuss the argument for locating assembly at the intersection between the lexicon and the syntax. Because of this decision, the semantic effect of syntactic arity operations will only be given (in section 5) as operations on denotation functions.

4.5 Examples

As a simple example, consider the derivation of unaccusative *open.*

(38) a. **Lexical entry for *open***

Event type: $\mathrm{P} = \lambda e\,\mathrm{open}(e)$

Arguments: $\mathrm{T} = \{[+\mathrm{c}+\mathrm{m}], [-\mathrm{c}-\mathrm{m}]\}$ (Agent, Theme)

Features: $\Phi = \{...\}$

We will show Φ as initially empty, since all features relevant to the Theta System are added during derivation. (Accordingly, we did not include features in (24), our model of the stored lexical entry.)

b. **Initial WS**

WS = <P, T, Φ>, a copy of the stored lexical entry.

c. **Derivation** (only changed components are shown)

1. From lexical entry: $T = \{[+c+m], [-c-m]\}$

2. Add ACC, MARK: $T = \{[+c+m], [-c-m]\}$; $\Phi = \{\text{ACC, MARK}\}$

3. Apply reduction: $T = \{[-c-m]\}$; $\Phi = \{\text{MARK}\}$

4. Merging instructions: MARK is present, so by merging instruction (32b), the surviving cluster projects internally (i.e., no EXT feature inserted).
$\Phi = \{\text{ACC}\}$

5. Assembly: *open* $+ \lambda x\, \lambda e\, \text{Theme}(e, x)$ =
$\textit{Proj}_0(\lambda x\, \lambda e\, \text{Theme}(e, x))(\lambda e\, \text{open}(e)) =$
$\lambda x\, \lambda e\, \text{open}(e) \wedge \text{Theme}(e, x)$

d. **Syntactic object**

denotation: $\lambda x\, \lambda e\, \text{open}(e) \wedge \text{Theme}(e, x)$

features: { ... } (none inserted by the Theta System)

Assembly produces a single denotation function and simultaneously clears the contents of P and Φ. The resulting syntactic object is ready to be used in syntactic derivation. It includes the assembled denotation function as the expression of its semantics, and consists of interpretable and uninterpretable features that encode, inter alia, the information that it projects as unaccusative. (In our model, this is expressed by the absence of the privative feature EXT.)

For a more complex example of the assembly operation, we illustrate the assembly of the final (highest) argument of the three-argument verb *peel,* whose lexical entry is given in (39a).

(39) a. **Lexical entry for *peel***

Event type: $\lambda e\, \text{peel}(e)$

Arguments: $\{[+c+m], [-c-m], [+c-m]\}$ (= Agent, Patient, Instrument)

Features: $\Phi = \{...\}$

b. **Initial WS**

WS = <P, T, Φ>.

c. **Derivation**

No arity operations.

1. Add features ACC and MARK. $\Phi = \{\text{ACC,MARK}\}$
2. Merging instructions: Introduce EXT feature, and order the arguments with [+c+m] last (highest). $\Phi = \{\text{ACC,MARK,EXT}\}$

d. **Partially built denotation**

(We suppress the details of how the first two arguments are assembled.)

$\lambda z\, \lambda y\, \lambda e\, (\text{peel}(e) \wedge \text{Instrument}(e, z) \wedge \text{Patient}(e, y))$.

Remaining argument: T = $\{[+\text{c}+\text{m}]\} = \{\ \lambda x\, \lambda e\, \text{Agent}(e, x)\ \}$

e. **Argument projection:** Add role [+c+m] to the partially built denotation of step *d.*

1. $\mathit{Proj}_2([+\text{c}+\text{m}])(\text{peel}(\text{Patient}, \text{Instrument})) =$
2. $\mathit{Proj}_2(\lambda x\, \lambda e\, \text{Agent}(e, x))(\lambda z\, \lambda y\, \lambda e\, (\text{peel}(e) \wedge \text{Instrument}(e, z) \wedge \text{Patient}(e, y))) =$
3. $\lambda z'\, [\mathit{Proj}_1(\lambda x\, \lambda e\, \text{Agent}(e, x))(\lambda y\, \lambda e\, (\text{peel}(e) \wedge \text{Instrument}(e, z') \wedge \text{Patient}(e, y)))] =$
4. $\lambda z'\, \lambda y'\, [\mathit{Proj}_0(\lambda x\, \lambda e\, \text{Agent}(e, x))(\lambda e\, (\text{peel}(e) \wedge \text{Instrument}(e, z') \wedge \text{Patient}(e, y')))] =$
5. $\lambda z'\, \lambda y'\, \lambda x\, \lambda e\, (\text{peel}(e) \wedge \text{Instrument}(e, z') \wedge \text{Patient}(e, y') \wedge \text{Agent}(e, x))$.

f. **Syntactic object:**

denotation = $\lambda z'\, \lambda y'\, \lambda x\, \lambda e\, (\text{peel}(e) \wedge \text{Instrument}(e, z') \wedge \text{Patient}(e, y') \wedge \text{Agent}(e, x))$

features = $\{$ACC,EXT,...$\}$

In this case the assembled word projects an external argument and carries an accusative Case feature, which must be checked during syntactic derivation.

4.6 When does assembly happen?

We now return to the following question: at what point is the information in the WS assembled into a denotation function? While I have located assembly just prior to lexical insertion, one might conceivably position it at any point between the stored lexical entry and the LF interface. We will consider, and reject, earlier or later alternatives.

1. **Early assembly:** Assembly is made at the earlest possible moment, before any arity operations on the verb entry.

2. **Late assembly:** Assembly is made somewhere in the course of syntactic derivation, after all syntactic arity operations.

The timing of the assembly step determines the form of semantic information at each stage of lexical-syntactic derivation: before assembly, arity operations (and other operations that manipulate the lexical item) must work on the structured representation we have defined; after assembly, they must

manipulate the semantic objects (functions) built by the assembly process. Conversely, the arity operations of the theta system must be defined so as to operate on the types of objects available to them. The non-redundancy principle (27) implies that we cannot have access to both kinds of structures at once: When the semantic form is assembled, the corresponding information is removed from our working copy of the lexical entry. We can summarize this as follows:

(40) **Post-assembly condition:** After assembly is complete, all arity operations must operate on the resulting semantic object; their semantic effect must be expressible as operators on denotation functions of our model-theoretic logic.

Early assembly would mean that a denotation function is built immediately, directly from the verb's lexical entry; all arity operations must work by manipulating the resulting function. This would allow us to define stored lexical entries that do not redundantly encode argument order, but obviously it would render useless our definitions of the arity operations in section 4.2, which operate on a modified copy of the lexical entry. As we have seen in section 3.4, however, some arity operations are impossible to define as manipulations of denotation functions. Early assembly would make it impossible to carry out these arity operations, and must therefore be rejected as an option.

Late assembly would most simply mean assembling the verb's denotation at the LF interface, in the course of computing the compositional interpretation of the entire sentence; but assembly might also happen at some intermediate point.[26] Adopting late assembly would require our model of syntactic derivation, and perhaps our semantics, to be adapted to include the WS. While this is a rather radical solution, it might not present insurmountable difficulties if it can be shown to be necessary—especially if we could assume such a syntactic model rather than actually developing one. The advantage of late assembly is that all arity operations manipulate the same sort of objects regardless of whether they operate in the lexicon or in the syntax. Consequently a lexical component rule, e.g., reflexivization, can be formally identical to its syntax counterpart. However, assembly at lexical insertion allows us to account for the differences between lexical and syntactic operations, since they apply to different kinds of objects.

[26] A possibility, proposed by Tanya Reinhart (personal communication), involves assembly at the VP level. Its potential advantages and disadvantages will not be pursued here.

5 Lexical and syntactic arity operations

In section 3.4 we saw that there are indeed arity operations that cannot be written as operations on denotation functions: Reduction (including decausativization), which completely removes a theta cluster from the verb's grid, and causativization, which replaces a cluster with its −c counterpart (as well as adding a new argument). Remarkably, these are exactly the arity operations that, according to the Theta System, are never encountered in the syntax. Saturation, arbitrarization and reflexivization are not restricted to the lexicon. They can occur in either component, although Reinhart and Siloni (2005) argue that the choice is not free but parametrically controlled by a parameter of UG (6). These operations are easy to define as operations on denotation functions, requiring only the manipulations already discussed in the conclusion of section 3.4.

Saturation and arbitrarization existentially close an argument, and can be readily applied in the syntax. For example, the operator defined in (41) will saturate the external argument of a transitive verb.

(41) $\text{SAT}_{ext}(V) = \lambda V_{<e,<e,st>>} \,.\, \lambda x\, \lambda e\, \exists y\, V(x)(y)(e)$ (Type: $<<\text{e}, <\text{e}, \text{st}>>, <\text{e}, \text{st}>>$)

Formula (42a) applies *SAT* to the transitive verb *break*. Part (b) shows the result of combining the passivized verb with an NP argument.

(42) a. $\text{SAT}_{ext}(\lambda x\, \lambda y\, \lambda e\, \text{break}(e) \wedge \text{Agent}(e, y) \wedge \text{Patient}(e, x)) =$

$\lambda x\, \lambda e\, \exists y\, \text{break}(e) \wedge \text{Agent}(e, y) \wedge \text{Patient}(e, x)$

b. The vase was broken. $\rightarrow$ $\lambda e\, \exists y\, \text{break}(e) \wedge \text{Agent}(e, y) \wedge \text{Patient}(e, \text{vase})$

Arbitrarization is similar, but the introduced variable is of a special sort marked for arbitrary human reference (see Chierchia (1995) for discussion of its semantics):

(43) $\text{ARB}_{ext}(V) = \lambda V_{<e,<e,st>>} \,.\, \lambda x\, \lambda e\, \exists y_{arb}\, V(x)(y)(e)$ (Type: $<<\text{e}, <\text{e}, \text{st}>>, <\text{e}, \text{st}>>$)

It is easy to write a family of operators that saturate the arguments of predicates of any arity.

Reflexivization can also apply in either the lexicon or in the syntax. It is straightforward to write reflexivization as an operator on formulas, given Reinhart and Siloni's (2005) conversion to the view that it involves identification of two roles (and not reduction of the internal role, as in the (Reinhart 2002) version of the theory).[27] An operator that reflexivizes a transitive verb can be written as follows:

[27] At the time, Reinhart (personal communication) raised a "conceptual problem" with the role identification proposal: If, as the Theta System proposes, individual theta features should be taken as having independent semantic content, what does it mean for an individual to be both Agent ([+c+m]) and

(44) $\text{REFL}(V) = \lambda V_{<e,<e,st>>} \,.\, \lambda x \, \lambda e \, V(x)(x)(e)$ (Type: <<e, <e, st>>, <e, st>>)

Following this approach, Reinhart and Siloni (2005) analyze lexical reflexivization as "bundling" two theta roles, i.e., identifying them with a single projected argument. Syntactic reflexivization, however, proceeds in a somewhat different way. Rather than directly bundle the two theta roles, it suppresses the Case feature of the verb. The corresponding theta role remains unassigned, and has the opportunity to be discharged only when the external theta role is merged. At that time, the two theta roles are bundled together and assigned to the external argument.

Explaining the distribution We conclude that assembly must take place after the lexical arity operations have applied, but may occur either before or after the syntactic operations. But we can go further than that: I propose that assembly in fact takes place at the dividing line between lexical and syntactic operations, just prior to lexical insertion and the beginning of syntactic derivation. We then have a way of correctly predicting which arity operations are restricted to the lexicon and which may occur in the syntax: causativization and decausativization are restricted to the lexicon *because* their operation cannot be expressed as an operator on functions. The remaining arity operations, which are not so restricted, can apply either in the syntax or in the lexicon.

Let us reiterate here that a suitably constructed model of the syntax could provide a work-around for these limitations (most simply, we could postpone or eliminate assembly and simply retain a WS-like structure throughout syntactic derivation); but the present proposal, which involves nothing more than what is necessary to assemble ordinary event semantics denotations out of an unordered set of theta clusters, correctly predicts the distribution of arity operations.

Patient ([−c−m])? We appear to be predicating of one and the same individual that it is a "sufficient condition" for bringing about some event e, and also that it is *not* a sufficient condition for bringing it about. But this would be a contradiction.

The problem disappears if we take features to characterize not an individual, but its relationship to an event. The feature +c says that an individual is related to an event in a way that makes it a sufficient condition for this event, while −c says that it is related in a way that does *not* make it a sufficient condition. There is no contradiction, only the entailment that our individual is related to the event in multiple ways; which is after all what assignment of multiple theta roles is intended to convey.

5.1 An alternative explanation

In the previous section I showed that an arity operation occurs in the syntax just if the manipulation it performs can be expressed as an operation on lambda forms; causativization and decausativization operations, which manipulate feature clusters in ways that cannot be so expressed, are limited to the lexicon. I have argued that these arity operations are so restricted precisely because syntactic arity operations must be expressible as operators on lambda forms.

Siloni (2002) proposes an alternative explanation of this distributional pattern: Noting that reduction and causativization both make changes in the number and identity of clusters in a verb's theta grid, she proposes that they are not possible syntactic operations because the theta grid of a predicate may not be changed in the syntax. This is summarized as the "lexicon interface guideline."

(45) **The Lexicon Interface Guideline** (Siloni 2002)

The syntactic component cannot change θ-grids: Elimination and modification of a θ-role as well as addition of a role to the θ-grid are illicit in syntax.

Reflexivization and saturation are understood as affecting the mapping of theta roles to syntactic arguments, not the inventory of theta roles themselves. Therefore they are permitted in the syntax.

Both explanations rule out deletions or modifications of feature clusters in the syntax. The account I have proposed follows from properties of the semantic framework and the proposed model, while Siloni's is based on the introduction of an intuitively plausible condition on what kinds of operations are permitted in the syntax. However well motivated, the Lexicon Interface Guideline was specifically introduced to account for the observed distribution of arity operations. In the absence of strong independent motivation, it must be considered a stipulation and therefore a less parsimonious explanation than the independently developed account I have proposed. On the other hand, the Guideline is a strictly syntax-internal explanation, a characteristic that might appeal to some readers.

The two accounts are conceptually compatible, since they adopt different means of explanation. If they made identical predictions, choosing between them would ultimately be a matter of taste; but Siloni's formulation makes a further prediction, which does not follow from our proposal: That it is also impossible to add a theta role in the syntax.

It appears that such operations do in fact occur in the syntax. The *applicative* construction is a well-known example of an operation that extends the argument structure of a verb, adding an internal

argument. It is found (inter alia) in many Bantu languages, where it is expressed by a derivational verbal suffix (usually *-i, -li,* or a cognate), and is extremely productive. In Kichaga, for example, "the new object may have the thematic roles of beneficiary, maleficiary, recipient, instrument, location, or motive (reason or purpose), depending on the semantics of the base verb." (Bresnan and Moshi 1990).

(46) Applicative: Kichaga (Bresnan and Moshi 1990)[28]

a. N-ā́-ī́-ly-à k-élyà.
Foc-1s-Pres-eat-FV 7-food

'He/She is eating food'

b. N-ā́-ī́-lyì-í-à m̀-kà k-élyà.
Foc-1s-Pres-eat-Appl-FV 1-wife 7-food

'He/She is eating food on (to the benefit/detriment of) his wife'

c. N-ā́-ī́-lyì-í-à mà-wòkő k-êlyâ.
Foc-1s-Pres-eat-Appl-FV 6-hand 7-food

'He/she is eating food with his/her hands'

That the added NP is an argument of the verb is not under dispute: It triggers object agreement on the verb, and it is treated as an object by subsequent productive arity operations, including passivization. If the applicative is a syntactic operation, as seems quite plausible given its productivity, regularity and other properties, then it contradicts the Lexicon Interface Guideline (45).

The system proposed in section 4 allows applicative formation to be formulated as an operation on denotation functions, and therefore predicts that it should be possible in the syntax.[29]

[28]Numeric prefixes in the gloss indicate noun class. 1S = class 1 (animate singular) subject agreement; APPL = applicative; FV = final vowel; FOC = focus; PRES = present tense.

[29]We have already seen formulas that add a theta role predicate to the denotation of the verb: that is just what the argument projection operators in (34) do. While the argument they insert is always higher (merges later) than existing arguments, it is just as easy to add an argument that merges first (or, indeed, in any intermediate position; cf. formula (23d)). The following formula will add a Beneficiary role to a transitive verb:

(i) $\text{APP}_2 = \lambda P_{<e,<e,st>>}\,[\lambda z\,\lambda x\,\lambda y\,\lambda e\,P(x)(y)(e) \wedge \text{Ben}(e,z)]$
(Type: $<<\text{e},<\text{e},\text{st}>>,<\text{e},<\text{e},<\text{e},\text{st}>>>>$)

6 Conclusion

I have outlined a concrete proposal for embedding the Theta System in a simple account of event semantics. By positing a change of representation at the time of lexical insertion, the proposal makes empirically correct predictions about the types of lexical operations that are possible in the lexicon and in the syntactic component. It must be remembered that the restrictions leading to these predictions are not dictated by conceptual necessity, but follow from properties of the formal framework (Montague-style truth-conditional semantics). If verb denotations are not really model-theoretic functions but some richer kind of entity, they might not be subject to the restrictions we have demonstrated. The proposed explanation, then, relies on the premise that the logical formulas commonly employed by semanticists are not just convenient notation, but have expressive content that corresponds to the formal power of the semantic operations that take place during lexical and syntactic derivation. In finding a correlation between properties of the formal framework and actual properties of arity operations, it can be argued that we have obtained empirical support for our choice of (broad) formal framework. The correlation can be taken as evidence that the semantic component of grammar does in fact have properties analogous to those of our truth-conditional, model-theoretic framework.

In the appendix that follows, we consider the details of what can and cannot be accomplished by the manipulations that our semantic framework makes available.

7 Appendix: Deleting a theta role from a lambda form

In this section I prove the claim, discussed in section 3.4, that there is no systematic way to completely remove a thematic role from the truth conditions of a verb denotation. Concretely, this means that reduction (employed by decausativization) and causativization cannot be defined as operations on functions, and therefore must be restricted to the lexical component.

Event semantics is designed to allow the participants in an event to be relatively independent. A proposition in "neo-Davidsonean" form, about an event and all its participants, is designed to license the inference of weaker propositions which do not mention some participants. For example, (47b) is logically entailed from (47a).

(47) a. Bill broke the glass.

$\exists e\,\text{break}(e) \wedge \text{Cause}(e, \text{Bill}) \wedge \text{Patient}(e, \text{glass})$

b. The glass broke.

$\exists e\,\text{break}(e) \wedge \text{Patient}(e, \text{glass})$

While such inferences are plainly valid in our logical system, I have asserted that the mechanics of lambda calculus (or any other system that treats verb denotations as indistinguishable from the corresponding truth-conditional functions) do not allow us to define a general mapping from the first type of predicate to the second. We cannot manipulate the denotation of the transitive form of *break,* for example, in such a way that we delete ("reduce") its Cause role.[30] That is, if we're given a verb denotation like *break*$^{(2)}$ (48a), there is no general way to construct from it a denotation like *break*$^{(1)}$ (48b), which is identical except that it does not mention a Cause participant. (The superscripts denote the arity of each form.)

(48) a. $\text{break}^{(2)} = \lambda x\,\lambda y\,\lambda e\,\text{break}(e) \wedge \text{Cause}(e, y) \wedge \text{Patient}(e, x)$

b. $\text{break}^{(1)} = \lambda x\,\lambda e\,\text{break}(e) \wedge \text{Patient}(e, x)$

More generally: Given a function f representing the conjunction of two or more predicates over eventualities, or a function from other types that leads to such a conjunction, there is no way to identify the predicates that were conjoined to create f—even if we know all but one of them. In the following sections, I provide a mathematical proof of this claim. As we will see, the problem cannot be solved unless fairly restrictive conditions are imposed; and if solvable, the solution cannot be formulated in a general way but must be expressed in relation to a particular model. The reason, in brief, is that the extension of f is not rich enough to tell us which event type predicate went into its definition. Although we can write verb denotations as combinations of predicates and logical operators in the language of lambda calculus, each denotation is simply a function; it is not possible to "look" inside it and recover the expressions that went into making it. For this reason, combining predicates together is a lot easier than pulling them apart.

Decausativization and causativization must additionally be able to detect the presence of particular roles: Decausativization can only reduce a [+c] role, while causativization must detect and adjust any existing +c roles to avoid conflict with the added Agent. This task raises similar problems,

[30]Note that the term *reduction* is used as defined in the Theta System: It is the complete elimination of a theta role from the meaning of a predicate. In lambda calculus, "reduction" refers to certain formal manipulations of lambda forms which can, of course, be successfully applied to verb denotations.

discussed in section 3.4. We do not consider them further in this appendix.

In section 3.4 we presented the analogy of a distance function, d, that gives us the straight-line distance of any geographical location from some reference point. If I use d to calculate the distance of some location from the reference point, and I tell you that it is 42195 meters, you do not have enough information to identify the location whose distance I calculated to arrive at this number: The problem cannot be solved because there are a great many points (a whole circle's worth) for which this distance is the answer. This corresponds to the first of our findings: If I create a verb denotation f by conjoining an event type predicate and one (or more) theta role predicates, the components are not identifiable: There are many combinations of predicates that would yield f. To be clear, it is not difficult to find a set of predicates that give f when combined; the insoluble problem is to find the predicates that *actually* went into building f.

Imagine now that I restrict the distance problem as follows: Instead of calculating the distance between arbitrary points, I restrict my attention to a specific set, perhaps the set of European cities (each represented by the location of its idealized center). In this case the problem of going from distances to cities *might* have a unique solution: This will be the case as long as no two cities have the same distance from our reference point. This corresponds to our second result: In section 7.4, I spell out the conditions that guarantee the existence of a unique answer for the operation of reduction, i.e., for going from a transitive verb denotation to the denotation of its reduced form.

Finally, consider what is involved in calculating the distance between two cities: Given the geographic coordinates of two locations, there is a mathematical equation (the so-called "orthodromic distance") that gives us their straight-line distance along the surface of the earth. But there could be no mathematical equation that will convert a distance into the name of a city, or even into its coordinates. This can only be done by checking the distance of each candidate city (perhaps using something like the table of driving distances found in road maps). This corresponds to our third and final result: Even when the conditions for uniqueness of the reduction operation are satisfied, the result can only be obtained by checking our function against the set V of all event type predicates in our model. In this respect, reduction (decausativization) is still substantively more complex than saturation or reflexivization, which are easy to express in closed form.

7.1 Minimal conditions

Recall (section 3.4) that saturation (existential closure) and reduction are different operations in the terminology of the Theta System, and yield predicates with distinct truth conditions. Formula (49b), the result of applying reduction to the verb *break,* can be satisfied by any event of breaking, whether or not this event has a Cause participant in our model; but (49c) can only be satisfied by events of breaking that do have a causal participant.

(49) a. *break* = $\lambda x\, \lambda y\, \lambda e$ break$(e) \wedge$ Cause$(e, y) \wedge$ Patient(e, x)

b. *break*$_{\mathrm{Red}}$ = $\lambda x\, \lambda e$ break$(e) \wedge$ Patient(e, x) (Reduction)

c. *break*$_{\mathrm{Sat}}$ = $\lambda x\, \lambda e\, \exists y$ break$(e) \wedge$ Cause$(e, y) \wedge$ Patient(e, x) (Saturation)

If we apply each of the reduced expressions to an individual v, we get the following predicates over events:

(50) a. λe .break$(e) \wedge$ Patient(e, z) (from reduced verb)

b. λe .$\exists y$ break$(e) \wedge$ Cause$(e, y) \wedge$ Patient(e, z) (from saturated verb)

Suppose our semantic model was such that all events of breaking happen to have a Cause participant. Knowing that e is an event of breaking would then allow us to conclude that there is some y that is its Cause. In other words, the two predicates above would be truth-conditionally equivalent since they would be satisfied by exactly the same set of events. Since reduction and saturation are in fact operations with truth-conditionally distinct results (representing unaccusatives and passives, respectively; cf. section 3.4), we will assume that this does not happen: We must allow that for some event types at least, there will be individual events with different assortments of roles.[31]

7.2 A simple version: Predicates over events

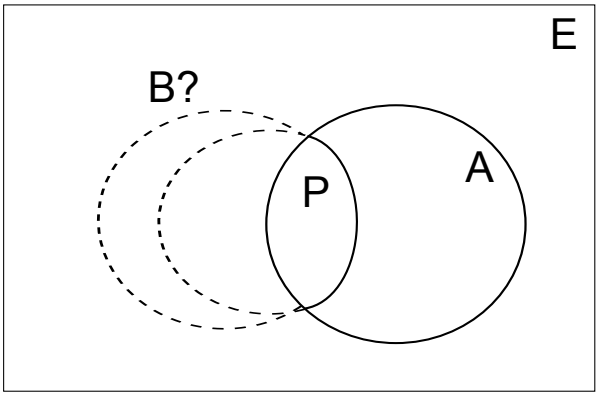
We begin with a lemma that may serve to persuade readers of the plausibility of the complete result:

(51) **Lemma** Let P and A be arbitrary subsets of a superset E, with $P \subseteq A$ and $A \neq E$. The equation $P = B \cap A$ does not have a unique solution for B.

[31] It is not necessary to assume that *all* event types show variation in their argument realization. Perhaps only a few event types do. Still, any definition of arity operations should be valid for all models and for all event types and theta roles to which the operations might be applied. We will assume for purposes of demonstration that we are dealing with the non-trivial cases.

Proof. Let X be any subset of $E - A$. Then $X \cup P$ is a solution, since $P = (X \cup P) \cap A$. Since $E - A$ is non-empty, there are multiple subsets and hence multiple solutions. □

The set relationships involved in the lemma are shown in the following diagram:



This diagram is meant to show that knowing the intersection of B with a known set is not enough information to determine B uniquely. E can be thought of as the set of events, A as the set of events that have an Agent (from now on we'll use Agent as our example theta role to be deleted), and P as the set of events denoted by the agentive version of some verb. Our task, which the lemma shows to be impossible, was to discover the non-agentive version. This task is impossible because of the lack of uniqueness: Because there are multiple sets that *might* have been used to construct P, it is impossible to determine the set that *was* used to construct P.

The condition $A \neq E$ is a weaker form of our assumption that some event types will allow variation in their inventory of theta roles: If A represents all events that have an Agent, and some event type comes in both agentive and non-agentive variants, we can minimally conclude that there exist some non-agentive events (not elements of A). This weaker condition, that some events are not elements of A, is all we needed to prove our lemma.

Knowing the intersection of B with A, then, is simply not enough to recover B: There are many sets whose intersection with A is P, and all are solutions to the equation $P = B \cap A$.[32]

[32] Note that switching to an intensional model does not help: if A, P and f are intensional functions from worlds w to sets of events such that $P_w = f(w) \cap A_w$, and $A_{w_0} \neq E$ for some world w_0, I can construct another solution f' with the property $P_w = f'(w) \cap A_w$, as follows: Choose an event $c \notin A_0$, and define f' as follows:

(i) $f'(w) = f(w)$ if $w \neq w_0$
(ii) $f'(w_0) = f(w_0) \cup \{c\}$ if $c \notin f(w_0)$
(iii) $f'(w_0) = f(w_0) - \{c\}$ if $c \in f(w_0)$

It is a trivial matter to apply this result to predicates over events:

(52) **Corollary** Let P and A be predicates over events, where $P(e) \Rightarrow A(e)$ and $A \neq E$ (i.e., there is some $e_0 \in E$ such that $\neg A(e_0)$). The equation

$$\forall e\ (P(e) \iff B(e) \wedge A(e))$$

does not have a unique solution for B.

Proof. Apply lemma (51) to the sets characterized by P, A and B. □

These results show that although we can write a lambda form as a conjunction of two predicates over events, appearances are deceiving: A characteristic function constructed as the conjunction of two predicates does not actually "contain" the conjoined predicates: it is just a function. There is no way to pick out the original conjuncts from among all the predicates that satisfy the equation of corollary (52). If we think of the conjunction of two functions as being itself a function (a mapping), this function is many-to-one and therefore has no inverse.

In the next section, we prove that the same holds when we move up from predicates over events to functions from individuals to such predicates. Removing a theta role is tantamount to going from an intersection to one of the intersected sets, which is impossible. Our proof must deal with more technicalities but otherwise follows the same approach.

7.3 Reducing a role from a transitive verb

Our goal is to prove the following:

(53) **Theorem:** Let $P_{<e,<e,st>>}$ be the denotation of a transitive verb, expressed as the conjunction of an event type predicate and two thematic roles, i.e., in the form

$$\lambda x\, \lambda y\, \lambda e\, .V(e) \wedge R_1(e, x) \wedge R_2(e, y).$$

Note that $c \notin P_0$, since $P_0 \subseteq A_0$. It is easy to see that $f \neq f'$, and that both are solutions to equation $P = B \cap A$ in all possible worlds.

There is no general way to compute the predicate $B_{<e,st>}$ that would have resulted by conjoining just the event type predicate and the internal (resp. external) thematic role of P.

It is easy to reverse the order of arguments in a lambda form, so the problems of deleting the internal or external role are equivalent.[33] Without loss of generality, we will consider reduction of the external argument. Moreover, our task assumes that we do not know which thematic role the external argument expresses (Agent, Experiencer, etc.). But since we are proving a negative result, it is enough to consider the easier task of deleting a known theta role, e.g., *Agent.* We will show that no solution to the simpler task is possible. From this, theorem (53) follows, since if we had a solution that works for an unknown theta role, we could use it even if we know that we are deleting an Agent role.

The conjunction of the internal theta role and the event type predicate (R_1 and V in the theorem's statement) does not need to be decomposed; we will treat their combination as a single predicate, $B_{<e,st>}$. For example, perhaps $B = \lambda x\, \lambda e\, \text{break}(e) \wedge \text{Patient}(e, x)$.

To prove theorem (53), then, it suffices to prove the following:

(54) **Theorem:** Given a function $P_{<e,<e,st>>}$ that is of the form $\lambda x\, \lambda y\, \lambda e\, .B(e, x) \wedge A(e, y)$, it is not possible to uniquely determine B if we only know P and *A*.

Proof. Suppose B can be determined given only P and A. Then B must be the unique set satisfying the condition of the theorem. We will construct another function that satisfies it, deriving a contradiction.

Without loss of generality, we can assume (as before) that reduction and saturation give truth-conditionally distinct results for our event type predicate; this happens just if there is some event e_0 such that $\exists x\, B(e_0, x) \wedge \nexists y\, A(e_0, y)$. (E.g., some events of the type in question are not agentive.)[34]

Let B be a solution to the equation of the theorem. Construct B' as follows:

i Fix an arbitrary individual $a \in D$.

ii Let $B'(e_0, a) = \neg B(e_0, a)$.

iii For all other e, x, let $B'(e, x) = B(e, x)$.

Suppose that B' is not a solution to the required equation. This means that when written into the equation, it gives us a function that differs from P. Then there must be some triple of arguments (an

[33] The expression $\lambda x\, \lambda y\, P(y)(x)$ is equivalent to P, except that its arguments are reversed.

[34] Since our theorem is concerned with computing an answer in the general case, it is enough to show that there are conditions where no answer exists. It follows that we cannot *always* compute an answer.

event and two ordinary individuals) for which B satisfies the equation of the theorem, but B' does not. By construction, B and B' are identical for all values not including e_0 as the event parameter, so the problem triple must include e_0.

Because e_0 has no A role, $A(e_0, y)$ is false for all y. It follows that $(B(e_0, x) \wedge A(e_0, y))$ is false for all x, y, and so is $B'(e_0, x) \wedge A(e_0, y)$. That is, if B satisfies the equation for some triple of values including e_0, so does B'. Hence B' satisfies the equation of the theorem for all triples of values, and therefore it is a solution.

We have shown that B and B' both satisfy the equation of the theorem but are distinct predicates, contrary to the assumption that B is unique. □

Since our problem does not have a unique solution, it is not possible to recover an event type predicate by removing (reducing) one theta role from a verb denotation: The conjoined denotation is compatible with a multitude of constituent predicates, with no way to determine which ones actually went into creating it.

7.4 Constraints that permit reduction

In our formulation of theorem (54), we have treated event type predicates as arbitrary predicates over events. But in our semantics, event type predicates are drawn from a set V, which is fixed by our model. This does not in itself invalidate our results (since we have not placed any restrictions on V, our predicates are still effectively arbitrary), but we now turn to the following question: Under what conditions on V, and on other aspects of the model, does the problem of reducing a theta role have a unique solution?

In this section, we show that uniqueness can be ensured in a linguistically plausible special case. Even in this restricted domain, however, reduction is qualitatively harder to express than operations, such as saturation, that do not alter or delete an existing theta role. In particular, the mapping from a transitive verb denotation to its reduced form cannot be expressed in the language of our semantic model; it can only be defined by reference to the metalanguage, or by stipulation ("brute force").

While there is no solution to the general problem of computing the reduction of a verb denotation P, the reduction operation in a linguistic context is not equivalent to looking for arbitrary predicates over events. In truth, the result of reducing one role of a transitive verb would be a

combination of a theta role and a basic event type; given the sets R and V of all possible theta roles and event types, respectively, we can construct a set $\mathcal{C}$ of all possible combinations. We can then simplify our task by considering only elements of $\mathcal{C}$ as possible solutions.

For simplicity, we once again ignore individual arguments and restrict ourselves to predicates over events. We assume that P is always of the form $P = B \cap A$ for some $B \in \mathcal{C}$, so that the task of finding B always has at least one solution. Informally, we will consider A to represent the set of agentive events. Our task then is to go from the agentive version of the verb to the non-agentive (reduced) version. We formalize this by defining a "solver" function Φ that gives us, for each set of agentive events P, all *qualifying* sets of events with the property that they have P as their agentive subset:

(55) $\Phi(P) \equiv \{X \in \mathcal{C} \mid P = X \cap A\}$

In effect, Φ is the closest we can come to a function that inverts the addition of an Agent. Our problem has a unique solution just if $\Phi(P)$ is a singleton for all non-empty P in its domain. ($\Phi(\emptyset)$ will give us all events whose type cannot have an Agent in our model; we can safely ignore these as irrelevant.) Note that Φ was specifically defined to allow us to reverse intersection with A, i.e., to delete an Agent theta role. Similar uniqueness conditions must hold for any other role that will need to be reduced.

Uniqueness is equivalent to requiring that if F and H are distinct event type predicates, and the event types they describe are compatible with Agents, the sets of *agentive* events of type F and H should be distinct. This is reasonable in a linguistic context: Even though there are agentive and non-agentive breaking events, I can expect agentive events of breaking to differ from agentive events of running, of sinking, etc. We do expect some overlap: If F and H are the sets of smashing and breaking events, respectively, there may be some events that qualify both as smashing and as breaking; but even then we can require that our models are rich enough for the agentive subsets of F and H to be distinguishable. In this linguistically plausible special case, $\Phi(S)$ is always a singleton if S is non-empty, and our problem at least has a solution: Given a set P of agentive events, there is only one event type predicate they could have come from.[35] We can now consider how difficult it is to find this solution.

We can express the solution to our problem as follows: $B = \Phi_0(P)$, where $\Phi_0(x)$ is the unique

[35] We have glossed over the effect of the remaining thematic role in this discussion.

element of the set returned by $\Phi(x)$. But we were only able to do this by referring to the set $\mathcal{C}$, and hence to the sets V and R, which are part of our model. Our solution function, in other words, is a contingent property of particular models meeting the distinctness condition we have outlined. By contrast, arity operations like saturation and reflexivization, which can apply in the syntax, can be expressed in a general, model-independent way: They are necessary properties of our semantic system.

It is also remarkable that all syntactic arity operations can be written as rather simple lambda forms: They are straightforward combinations of the elementary operations of standard model-theoretic logic. In a sense, they can be said to be logically simpler operations than deleting a role. To demonstrate this, consider that all the predicates and logical operators we have used in our semantics represent total, not partial functions.[36] The composition of total functions is also a total function, and thus every expression we can construct in our logical language will be defined (have a value) over all objects of the appropriate type. The solver function $\Phi_0(P)$, on the other hand, is only defined in models, and for event types, for which $\Phi(P)$ is a singleton. It follows that Φ_0 cannot be written as an expression in our logical calculus; since it is a partial function, it cannot even be introduced as a primitive unless we allow partial functions in our semantic system.

If we extend our logic with the ι (iota) operator for higher types, we can now express Φ_0 as a logical formula: $\Phi_0(P) = \iota Q \,.\, (P = Q \cap A)$.[37] Iota, of course, defines a partial function.

In short, saturation and other arity operations that apply in the syntax can be expressed in a very restrictive logical system, without partial functions. While the iota operator is commonly utilized in natural language semantics (at least for simple types), it can still be argued that a mapping that irreducibly requires such powerful tools is logically more complex, and perhaps cognitively more difficult to carry out, than the much simpler operations of adding a thematic role.

Summary We have shown that (a) the problem of deleting a theta role does not have a solution (because it does not have a unique one) unless certain uniqueness properties hold of the inventory of event types, V. (b) If the uniqueness condition holds, the mapping representing deletion of a theta role is not be expressible in a logical language that allows only total functions. (c) Deletion of a theta role

[36] While Landman (2000) defines theta roles as partial functions from events, we have defined them as truth-valued relations between events and individuals.

[37] $\iota x \, P(x)$ reads "the unique x with the property $P(x)$," so our formula simply states that $\Phi_0(P)$ is the unique solution to our equation.

can be expressed in a logical language that includes the iota operator. It is then contingent on the necessary uniqueness properties.

References

Adger, David. 2000. *Core syntax: A minimalist approach*. Oxford: Oxford University Press.

Baker, Mark. 1988. *Incorporation: A theory of grammatical function changing*. The U. of Chicago Press.

Bresnan, Joan, and Lioba Moshi. 1990. Object asymmetries in comparative Bantu syntax. *Linguistic Inquiry* 21:147–185.

Chierchia, Gennaro. 1989/2004. A semantics for unaccusatives and its syntactic consequences. In *The unaccusativity puzzle*, ed. Artemis Alexiadou, Elena Anagnostopoulou, and Martin Everaert. Oxford University Press. (Circulated as ms., Cornell University, 1989).

Chierchia, Gennaro. 1995. The variability of impersonal subjects. In *Quantification in natural languages*, ed. Emmon Bach, Eloise Jelinek, Angelika Kratzer, and Barbara H. Partee, volume 1, 107–143. Dordrecht: Kluwer.

Chomsky, Noam. 1981. *Lectures on government and binding*. Berlin: Mouton de Gruyter, seventh edition. Originally published by Foris. This edition published 1993.

Chomsky, Noam. 1995. *The minimalist program*. Cambridge, MA: MIT Press.

Davidson, Donald. 1967. The logical form of action sentences. In *The logic of decision and action*, ed. Nicholas Rescher. Pittsburgh: University of Pittsburgh Press.

Di Sciullo, Anne Marie, and Edwin Williams. 1987. *On the definition of word*. Linguistic Inquiry Monograph 14. Cambridge, MA: MIT Press.

Dimitriadis, Alexis. 2004. An event semantics for the Theta System. Ms., Utrecht institute of Linguistics OTS, March 2004.

Grimshaw, Jane. 1990. *Argument structure*. Cambridge, MA: MIT Press.

Halle, Morris, and Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In *The view from Building 20*, ed. Kenneth Hale and Samuel J. Keyser, 111–176. Cambridge, MA: MIT Press.

Hornstein, Norbert, Jairo Nunes, and Kleanthes K. Grohmann. 2005. *Understanding minimalism*.

Cambridge University Press.

Horvath, Julia, and Tal Siloni. 2003. Against the *little-v* hypothesis. Paper presented at the Incontro de Grammatica Generativa, Urbino, Tel-Aviv University, February 2003.

Horvath, Julia, and Tal Siloni. Forthcoming. The thematic phase and the architecture of grammar. In *Concepts, syntax and their interface*, ed. Martin Everaert, Marijana Marelj, Eric Reuland, and Tal Siloni. Cambridge, MA: MIT Press.

Horvath, Julia, and Tal Siloni. To appear. Causatives across components. *Natural Language and Linguistic Theory* .

Kratzer, Angelika. 1996. Severing the external argument from its verb. In *Phrase structure and the lexicon*, ed. Johan Rooryck and Laurie Zaring, 109–137. Dordrecht: Kluwer.

Krifka, Manfred. 1989. Nominal reference, temporal constitution and quantification in event semantics. In *Semantics and contextual expression*, ed. Renate Bartsch, Johan van Benthem, and Peter van Emde Boas. Dordrecht: Foris.

Krifka, Manfred. 1992. Thematic relations as links between nominal reference and temporal constitution. In *Lexical matters*, ed. Ivan A. Sag and Anna Szabolcsi, 29–53. Stanford: CSLI.

Landman, Fred. 2000. *Events and plurality: The Jerusalem lectures*. Dordrecht: Kluwer.

Link, Godehard. 1998. *Algebraic semantics in language and philosophy*. Stanford: CSLI Publications.

Manzini, Rita M. 1983. On control and control theory. *Linguistic Inquiry* 14:421–446.

Marantz, Alec. 1984. *On the nature of grammatical relations*. Linguistic Inquiry Monograph 10. Cambridge, MA: MIT Press.

Marantz, Alec. 1997. No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. *University of Pennsylvania Working Papers in Linguistics* 4:201–225.

Marelj, Marijana. 2002. Rules that govern the cooccurrence of theta clusters in the 'theta-system'. *Theoretical Linguistics* 28:357–373.

Marelj, Marijana. 2004. Middles and argument structure across languages. Doctoral Dissertation, Utrecht University.

Neeleman, Ad, and Hans van de Koot. This volume. The linguistic expression of causation.

Parsons, Terence. 1990. *Events in the semantics of English: A study in subatomic semantics*. Cambridge, MA: MIT Press.

Perlmutter, David, and Paul M. Postal. 1984. The 1-advancement exclusiveness law. In *Studies in*

relational grammar, ed. David M. Perlmutter and Carol Rosen, volume 2, 81–125. Chicago: The U. of Chicago Press.

Pesetsky, David. 1995. *Zero syntax: Experiencers and cascades*. Cambridge, MA: MIT Press.

Potashnik, Joseph. This volume. Emission verbs.

Rákosi, György. 2006. Dative experiencer eredicates in Hungarian. Doctoral Dissertation, Utrecht University.

Reinhart, Tanya. 2000. The theta system: Syntactic realization of verbal concepts. OTS Working Papers in Linguistics 00,01/TL, Utrecht institute of Linguistics OTS.

Reinhart, Tanya. 2002. The theta system: An overview. *Theoretical Linguistics* 28:229–290.

Reinhart, Tanya. Forthcoming. The theta system: Unaccusative and experiencer derivations. In *Concepts, syntax and their interface*, ed. Martin Everaert, Marijana Marelj, Eric Reuland, and Tal Siloni. Cambridge, MA: MIT Press. Edited for publication by Tal Siloni.

Reinhart, Tanya, and Tal Siloni. 2004. Against an unaccusative analysis of reflexives. In *The unaccusativity puzzle: Explorations of the syntax-lexicon interface*. Oxford University Press.

Reinhart, Tanya, and Tal Siloni. 2005. The lexicon-syntax parameter: Reflexivization and other arity operations. *Linguistic Inquiry* 36:389–436.

Siloni, Tal. 2002. Active lexicon. *Theoretical Linguistics* 28:383–400.

Siloni, Tal. 2008. The syntax of reciprocal verbs: An overview. In *Reciprocals and reflexives: Cross-linguistic and theoretical explorations*, ed. Ekkehard König and Volker Gast. Berlin: Mouton de Gruyter.

Williams, Edwin. This volume. Combine.