An event semantics for the Theta System

Alexis Dimitriadis
Utrecht institute of Linguistics OTS

March 11, 2004

DRAFT — COMMENTS AND CORRECTIONS WELCOME

1 Introduction

The theory of the Theta System (Reinhart 2000, 2002) gives a compositional account of verbal argument structure and argument structure alternations. Although it makes concrete predictions with respect to argument projection and syntax, Reinhart has not provided a definition of the semantics, and semantic operations, associated with the system's primitive elements and operators. In this paper I provide a semantic implementation of the system's components, based on a straightforward embedding of the Theta System in the "event semantics" of Davidson (1967), Parsons (1990). While it has not been my intent to extend or "explain" the theta system, the design proposed in these pages turns out to have empirical consequences; these are explored in section 5.

The Theta System does not consider the order of thematic arguments to be specified in the lexical entry; it is determined after the operation of lexical marking and argument-structure operations, according to the *CS merging instructions.* But syntactic constituency as well as lambda formulas involve argument positions that must be saturated in a particular order. In order to end up with the usual kind of semantic object, a function expressible as a formula of lambda calculus, there must be a *change in representation* at some point between the lexical entry and the interpretation of the sentence at the LF interface. Before the change-over, which I will refer to as *translation,* 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: formulas of intensional logic for the semantics, and syntactic features for the syntax. I will show that translation takes place after the application of lexical-component arity operations, but before syntactic operations; it is simplest to locate it just before insertion of lexical items in the numeration.

This architecture, adopted as a natural solution to the practical problem of designing a semantics of the theta system, turns out to have desirable empirical consequences: It accounts, in a natural way, for certain well-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, other arity operations are cross-linguistically restricted to the lexicon. A number of known differences between the lexicon and the syntax can be brought to bear on the question of why some operations can apply in either domain while others cannot, but until now there has been no comprehensive explanation.

In formalizing both types of arity operations within the semantic framework I will propose below, it turns out that the arity operations that can apply in the syntax are exactly those that can be expressed as basic operations on logical formulas: Namely, an operator on formulas can existentially close off a role or identify two roles; but it cannot simply delete the first (or n-th) theta role (which would be tantamount to deleting one of the conjuncts of a logical formula), nor replace it with another. On the other hand, any of

This work was inspired by Tanya Reinhart's seminars on the Theta System in 2001 and 2002. I am grateful to Tanya Reinhart, Marijana Marelj, Tali Siloni, Maribel Romero, and Danny Fox for discussion, suggestions, questions, and answers that have contributed to the direction and substance of this work.

This paper is a revision of a draft circulated in 2002. Earlier versions have been presented at the 26th GLOW Colloquium in Lund, Sweden, at the Brown University Workshop on Direct Compositionality (2003), and to audiences at the Utrecht institute of Linguistics.

these manipulations can be carried out in a suitably-structured system of our own design. Indeed, we find that operations that require existential closure of an argument position or identification of two argument positions (Saturation, Arbitrarization, Reflexivization) can occur in the syntax, while operations that require outright deletion of a theta role, or substitution of one thematic role for another (Reduction, Causativization) only occur in the lexicon.

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 stated as a general rule that manipulates the semantic encoding available during syntactic derivation. In the lexicon a different, more articulated encoding is used, which allows all of the arity operations to take place. While it could be argued in response that we are dealing simply with accidental consequences of an arbitrary system, the encoding I have assumed for the syntactic component is simply the usual view of meanings as functions in some intensional calculus, expressible as lambda terms. Although any system can be elaborated in such a way that it allows arbitrary transformations on its objects, the natural (and standard) way to encode the semantics of verbs simply does not provide enough structure to allow certain manipulations. Since these manipulations are exactly the ones that are unattested in the syntactic component, their distribution confirms the view that the expressive power of the standard semantic framework is approximately right for a semantics of natural language.

The paper is organized as follows: In the next section, I provide an overview of the theory of the theta system and some of its more relevant properties. Section 3 discusses the embedding of event semantics in the framework of the theta system, and defends the conclusion that there is a change in representation between the lexicon and eventual semantic interpretation. 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 expression of the various arity operations in each component.

2 The Theta System

The Theta System framework provides a compositional treatment 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 argument structure operations 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 in 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.* 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 apply *after* any argument structure operations such as the passive.

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 unaccusative formation in various languages are examples. These operations rely on a very restricted set of universal *arity operations,* each of which performs a particular manipulation of theta grids. Arity operations include internal and external *Reduction* (deletion of a theta role), *Saturation* (existential closure of a role), and *Causativization* (which transforms a verb's theta grid, adding an agent and "decausativizing" the original agent or cause).

Argument structure operations take place in two domains, the *computational lexicon* and the *(morpho)syntax.* There is some redundancy in the system: Several arity operations (e.g., reflexivization) are known to apply either in the (computational) lexicon or in the syntax, depending on the language. Others, such as Causativization, are restricted to the lexicon. Neither the reasons for this redundancy nor the restriction of certain operations to one or the other component are well understood at present; as mentioned already, the present paper attempts to shed some light on this issue.

Aside from this issue, it is not my goal here to "explain" the properties of the Theta System; the present proposal merely extends the original formulation of the Theta System, which is primarily concerned with modeling syntactic phenomena, to state (some) semantic components of the entities and operations of the system. On the other hand, I will deviate in some details from the technical devices presented in the Theta System. These deviations, made for reasons of simplicity and sometimes of personal taste, should not have any empirical consequences.

I do not attempt to provide a full exposition of the Theta System, nor (especially) of the evidence for its claims; although I reproduce arguments for some issues central to the findings of section 5.2, for the most part I provide only an overview of its workings. The reader is referred to Reinhart (2000, 2002) for the details.

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. Of the nine possible combinations, eight play a role in the theory; the empty cluster is presumed to be non-occurring.

The following summary is based on Reinhart (2002).

(2) **The feature clusters**

[+c+m]	Agent
[+c−m]	Instrument, other??
[−c+m]	Experiencer
[−c−m]	Theme / Patient
[+c]	Cause (Unspecified for *m*; consistent with Agent or Instrument)
[+m]	"Special experiencer": the subject of verbs like *love, know, believe* in English, which always generate externally. Perhaps also animate subjects not involving necessary agency, e.g., of *laugh, cry, sleep.*
[−c]	Goal, Benefactor, and similar internal roles, typically dative or PP.
[−m]	Subject Matter, Locative Source (Unspecified for *c*).

To illustrate, a [+c] argument expresses any causal role; a subject associated with this role may be interpreted as an unspecified Cause, an Agent, or an Instrument, while a [+c+m] subject can only be an Agent. This is Reinhart's account of the difference between the verbs in (3) and those in (4); the former allow a variety of causally involved subjects (including inanimate causes and instruments), while the latter can only be construed with an agent.

(3) [+c] **verbs**
  a. The wind/Max/the key opened the door.
  b. The storm/Max/the stone broke the window.

(4) [+c+m] **verbs**
  a. The baby/*the spoon/*hunger ate the soup.
  b. Lucie/*the razor/*the heat shaved Max.
  c. Lucie/*the cold/*the desire to feel warm dressed Max.

As with any system of theta roles, determining the 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 theta clusters corresponding to the standard theta roles (such as Agent) can generally be assigned on the basis of their semantics, while the presence of other roles like [−m] must often be inferred from a verb's argument structure alternations, in part relying on the predictions of the theory.

An interesting issue is the relationship between theta clusters and the traditional theta roles. For example, consider the roles Goal and Benefactor, both of which correspond to the [−c] cluster. Do we have two distinct theta roles characterized by the same value for a certain pair of features, or do we just have two names for the same thing? While Reinhart does not explicitly address the question, she does stress that feature clusters have semantic content, and in her writings the feature clusters *are* the theta roles. Nevertheless her system is designed to be an adequate account of argument structure alternations, not of regularities in meaning, and it is possible that the eight cluster types she postulates are too few for a complete semantic account of thematic roles, making a two-layer architecture necessary.[1]

The question is difficult to decide, not least because it is not at all clear which entailments belong with a system of general-purpose theta roles and which should follow from the meaning of the verb itself. For simplicity and ease of exposition, I will assume that feature clusters *are* the theta roles. This is almost certainly not adequate if we adopt Marelj's (2002) system of disambiguating underspecified 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 that are relevant to syntactic projection, too few to capture the diversity of theta role semantics. But I consider it an open question whether the full system of eight clusters is adequate as a complete system of semantic roles.

2.2 Argument projection and linearization

As already mentioned, the lexical entries for 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." Arguments are not ordered according to a universal, 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:

(5) **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], mark the verb with the ACC feature.

(6) **CS merging instructions:**[2]
   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 accusative case to assign (Acc feature) and no other argument is available to receive it.

The marking rules in (5) are explicitly restricted to underlyingly transitive entries. Underlyingly intransitive verbs are not marked, and consequently the merging instructions always cause them to project as

[1] This sort of arrangement is suggested as a possibility by (Chomsky 1981:139).
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, semantically relevant, conception of theta roles.

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

unergative.

Note also that the external argument is projected by the verb, not contributed extrinsically (e.g., by a syntactic head, as proposed by Kratzer (1996) and others).[3] By treating internal and external arguments alike, Reinhart is able to explain alternations such as the following, in which one argument (the Experiencer, [−c+m]) may project either internally or externally, depending on the other arguments of the verb.[4]

(7) a. The letter worried John.
    b. John worried.

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

The Theta System derives the function of grammatical operations that manipulate argument structure, such as the passive, from a small set of universal *arity operations* that manipulate theta clusters.

Some arity operations, such as reflexivization, are known to apply either in the (computational) lexicon or in the syntax, depending on the language. Other operations, such as Causativization, are restricted to the lexicon.[5] Diagnostics for the distinction include productivity (syntactic languages have productive reflexivization) and reflexivization into ECM complements (possible in syntactic languages only). French and Italian are syntactic, while English and Hebrew are lexical. (Reinhart and Siloni, forthcoming). Thus the grammatical French example (8a), illustrating coreference between the matrix subject and the subject of an ECM complement, contrasts with its ungrammatical Hebrew counterpart (b). The evidence is presented in more detail in section 2.5. In section 5.2, I argue that the division between the two types of operations follows from necessary characteristics of the implementation proposed below.

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

The inventory of arity operations proposed by Reinhart is presented in summary form below.

(9) **Some properties of arity operations**
    a. **Saturation** is the operation that applies (inter alia) in passive formation. It existentially closes an argument, which is consequently present for semantic interpretation, but is normally not realized syntactically. (It can be re-introduced as an oblique, or referred to by other constructions).
       Arbitrarization is a variant of saturation that involves existential closure by a variable marked for "arbitrary human" reference (Chierchia 1995).[6]
    b. **Reduction** completely eliminates an argument role. It completely removes this role from the verb's argument grid. Reduction can only apply in the lexicon.

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

[4]It can be shown that the verb *worry* in (7b) is not unaccusative. See section 2.4 for the details of Reinhart's account.

[5]Besides the arity operation Causativization, there is a syntactic "causativization" construction that, as Reinhart (2002) argues, has quite different properties.

[6]Arbitrarization is not in the original inventory of arity operations for the Theta System; its introduction was proposed by Marijana Marelj.

External reduction, or **Expletivization,**[7] can only apply to a multi-argument entry, one of whose roles is a [+c] cluster. That argument is removed (completely).
Reduction is responsible, inter alia, for deriving unaccusatives.

c. **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.[8]

d. **Causativization** (Expansion) *adds* a role: it subtracts the +c feature from some argument of the verb (the original agent or cause), if there is one, and adds a new [+c+m], i.e., Agent, argument. It can only apply in the lexicon.[9]

Word derivation in the Theta System follows a sequential model. Saturation and reduction apply after assignment of Case and internal-external indices ("lexicon marking"); in addition to their effect on theta clusters, they eliminate the Accusative feature of the verb (fully or partially), if it is present. But causativization applies before marking, and therefore influences argument projection and Accusative assignment.

Reinhart states a number of other conditions, generally empirically motivated, on arity operations. Among these is the unsurprising fact that saturation and reduction are mutually exclusive: they cannot both apply to the same lexical item.

2.4 Unaccusatives in the Theta System

To illustrate the system in operation, we new consider Reinhart's treatment of unaccusatives. Following Chierchia (1989) and Levin and Rappaport Hovav (1995), Reinhart derives causative/unaccusative pairs not through causativization of an intransitive, but in the opposite direction. The transitive form is taken to be basic, and the unaccusative form is derived by reduction.

(10) open([+c], [−c−m]) → open([−c−m])
a. Mary opened the door.
b. The door opened. (unaccusative)

Thus unaccusative (10b) is derived from (a) via the external reduction operation shown above. The same reduction operation can apply to the verb *worry*, as shown in (11). The intransitive alternate of *worry,* however, is unergative rather than unaccusative. This is due to other properties of the system: The CS merging rules allow the remaining argument of *worry* to project externally, while that of *open* must project internally.

(11) worry([+c], [−c+m]) → worry([−c+m])[10]
a. The letter worried John.
b. John worried. (unergative)

In particular, the argument of unaccusative *open* is a [+c+m] (Patient) cluster. The Lexicon Marking rules mark it with the index *2,* which forces it to merge internally, yielding an unaccusative. But *worry* takes a [−c+m] (Experiencer) argument, which is not marked by the Lexicon Marking rules; therefore it can merge either internally or externally. Since Reduction also eliminates the Accusative assignment feature, the experiencer of *worry* is free to merge externally (and therefore *must* merge externally), yielding an unergative.

[7]Not to be confused with the "Expletivization" operation defined by Chierchia (1989), which *adds* a semantically null external argument to the verb.

[8]This description reflects Reinhart and Siloni's (2003) current analysis. In earlier treatments, Reflexivization is analyzed as internal Reduction.

[9]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 some such analysis (e.g., capturing the insights of Dowty 1976) is compatible with the system, but I have nothing to add to this subject.

[10]Reinhart actually analyzes the verb *worry* as having a third, [−m] (Subject Matter) argument (following Pesetsky 1995). This argument is not a involved in the matters discussed here.

Recall that transitive *open* is known to have a Cause subject because inanimate causes and instruments (such as *the wind* or *the key*) are possible subjects. On the other hand, the transitive alternate of *walk* must subcategorize for an Agent, since *hunger* (a cause) or *the leash* (an instrument) are unacceptable as subjects:

(12) Max/*hunger/*the leash walked the dog to the food bowl.

Expletivization can only remove a Cause cluster, never an Agent. It follows that pairs like the following, which involve an Agent, cannot be derived in the same manner. They must be derived by *adding* an argument to the intransitive form, through *Causativization.*

(13) a. They ran/walked → She ran/walked them.
b. They worked hard → She worked them hard.

(14) Danny axal bananot. → Aba xe'exil et Danny bananot. (Hebrew)
Danny ate bananas Daddy fed (Acc) Danny bananas

Such verbs are never unaccusative. The system captures this fact by limiting application of the Lexicon Marking rules to entries that are transitive at the point when the rules are applied. The sole argument of underived (root) intransitives will never be indexed and therefore will always project externally, regardless of thematic role.

2.5 Lexical vs. syntactic operations

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

In particular, the theta system assumes that certain arity operations apply only in the lexicon, while others can apply either in the lexicon or in the syntax. Moreover, some arity operations, including Reflexivization, apply in the lexicon in some languages and in the syntax in others. For example, Reflexivization takes place in the syntax in French and Italian (which are therefore referred to as "syntactic" languages), but in the lexicon in Hebrew and English ("lexical" languages).

Reinhart considers this to be controlled by a parameter of Universal Grammar:

(15) **The Lex-Syn Parameter:** (Reinhart and Siloni 2003)
UG allows thematic arity operations to apply in the lexicon or in syntax.

Allowing arity operations to apply in two domains is less economical than restricting them to a single domain, but Reinhart and Siloni argue that the distinction accounts for a large number of empirical regularities cross-linguistically. While I will not attempt to defend the theta system's view of the grammar, I will consider the case of Reflexivization in more detail in order to provide some illustration of its workings, and some explication of just what "syntactic" and "lexical" derivation involves.

2.5.1 Two types of reflexivization

As shown already by Kayne (1975), the Romance clitic *se* is not an anaphor but an intransitivizing verb modifier. Reinhart and Siloni (2003, forthcoming) argue that it is a reflex of an arity operation, *Reflexivization,* that takes place in the syntax. (This operation is unrelated to the use of anaphors like *himself,* which build reflexive constructions without modifying the argument structure of the verb).

Reflexivization is syntactic in French, Italian, German, and Serbo-Croatian (inter alia), but takes place in the lexicon in Hebrew, English, Dutch, Russian and Hungarian. The basis for this claim is a cluster of several notionally independent properties, which covary. Reinhart and Siloni argue that they are consequences of the Lexicon-Syntax parameter. Some of the properties they identify are as follows:

1. **Productivity** Syntactic reflexivization is productive. Lexical reflexivization is generally restricted, sometimes to a handful of verbs (e.g., in Russian).

2. **Ambiguity** The reflexive affix in the languages under consideration is ambiguous between reflexive and reciprocal functions. However, in lexical languages there are few if any verbs whose reflexivized form is ambiguous between reflexive and reciprocal meanings; a verb can either become reflexive or reciprocal (or neither), but not both. In syntactic languages, on the other hand, such ambiguity is common. For example, in German *sich schlugen* can mean 'to hit oneself' or 'to hit each other'.

3. Reflexivization across **Exceptional-Case-Marking constructions** As Marantz (1984) pointed out, a lexical operation could only identify coarguments of the same lexical item; therefore the well-formedness of example (16a) cannot be explained by a lexical account. On the other hand, such examples are ill-formed in lexical languages, as example (b) shows.

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

4. **Nominalizations** Reflexivization in lexical languages can feed nominalization, which takes place in the lexicon.

(17) a. *hitraxcut* 'self-washing' (Hebrew; lexical)
b. *mos-akod-ás* 'self-washing' (Hungarian; lexical)
wash-Refl-Nom
c. (Nothing of the sort in languages with syntactic reflexives)

5. Reflexive interpretation of **agent nominals** This is only possible in lexical languages, such as English:

(18) a. She dresses slowly because she is an elegant dresser.
b. Jean est un excellent habilleur/maquilleur
Jean is an excellent dresser/make-up-er (of others only)

6. **Accusative assignment** While both kinds of reflexivization are intransitivizing operations, and withdraw accusative case from the reflexivized argument, lexical languages withdraw accusative Case even when a dative argument is being reflexivized. Syntactic languages can reflexivize a benefactive or indirect object, leaving the direct object intact.

(19) a. Jean s'est acheté une voiture. (French; syntactic)
'Jean bought himself a car'[11]
b. Jean s'est envoyé une lettre.
'Jean sent a letter to himself'

(20) * Dan hištale'ax mixtav. (Hebrew; lexical)
Dan sent(Refl) letter

7. **Discontinuous reciprocals** A number of additional properties distinguish *reciprocal* verbs (also formed via Reflexivization). Among them is the ability to form the *discontinuous reciprocal* construction (Siloni 2001, Dimitriadis 2002), shown in the (b) sentences below. This construction is only possible in lexical languages.

(21) a. János és Mari csókol-óz-t-ak. (Hungarian; lexical)
János and Mari kiss-Rcp-Pst-3pl
'Janos and Mari kissed'

[11] The English translation involves reflexivization of the benefactive argument by means of a reflexive pronoun (not a reflexive verb); therefore its grammaticality is not relevant to the construction under discussion.

b. János csókol-óz-ott Mari-val.
   János kiss-Rcp-Pst Mari-with

(22) a. John and Mary collided. (English; lexical)
   b. John collided with Mary.

(23) a. Giovanni e Maria si sono abbracciati. (Italian; syntactic)
   Giovanni and Maria SI are hugged
   b. * Giovanni si è abbracciato con Maria.
   Giovanni SI is hugged with Maria

Some of the above properties are clearly related to the question of the locus of reflexivization (binding in ECM constructions, feeding nominalizations). For others the relationship needs to be explained, but is nevertheless empirically present.

2.6 The Lex-Syn parameter revisited

While the syntax-lexicon distinction can account for a cluster of correlated properties in these languages, there are also exceptions. As Siloni (2002) points out, "syntactic" languages still have isolated cases of "lexicalized" reciprocals, which show the characteristics of lexical reciprocals. In French, a syntactic language, a few reciprocal verbs nevertheless allow the "discontinuous reciprocal" construction, which is considered characteristic of lexical languages only.

(24) a. battre 'beat'
   b. se battre = to quarrel (lexically derived)
   c. se battre = to beat each other (syntactically derived)

(25) Jean se bat avec Marie.
   a. 'Jean quarrels with Marie.'
   b. * 'Jean and Marie beat each other.'

But do the lexical reciprocals of syntactic languages differ from those of lexical languages? It seems that they do not. A lexical language has some number of lexical reciprocals, with particular identifying properties. A syntactic language has a large number of productive, syntactically derived reciprocals, and a number of frozen lexical forms; in all respects, the latter resemble the reciprocals of lexical languages.

So all languages, in principle, allow lexical reciprocals. If the case of reciprocals is representative of other manifestations of the Lex-Syn parameter, this parameter is not really about where arity operations apply, but about what's possible in the syntax. It might be reformulated as follows:

(26) **Lex-Syn Parameter (revised):**
   A language may allow or prohibit arity operations in the syntax.

Lexical reflexivization is universally allowed. Even in lexical languages, we sometimes find just a handful of reflexive verbs (e.g., in Russian). It is therefore not surprising that lexical reflexives in syntactic languages are few and far between.

3 Event semantics and the theta system

3.1 Desiderata

We begin by making explicit some of the desired properties that a satisfactory analysis must reflect. While there will not be universal agreement on all of these goals, it is hoped that readers will find themselves generally sympathetic to most of them. The analyses advocated in the rest of the paper attempt to provide natural solutions to the concerns presented here.

1. **Properties of the Theta System**

a. The theta roles assigned to arguments are not purely formal features: they have semantic content.
b. The ordering of arguments is constrained, but underdetermined, by general principles of the theta system.
c. There are general co-occurrence restrictions on theta clusters.
d. Individual verbs *may* show lexically-specified deviations from the pattern predicted by the general rules of the theta system.

2. **Properties of the semantic framework**

a. The account should be compositional: It should consist of semantic representations for the elementary parts of the system, along with a general way of deriving the semantics of compound objects from those of their parts.
b. The account should be local: The semantic material associated with a verb must be inserted at only one place in the derivation (below a V^0 in a syntactic tree).[12]
c. The semantic effects of argument structure operations (e.g., reflexivization) should be expressible by general formulas operating on the representation of the verb entry.
d. The semantic content of theta roles must be explicitly represented.

3. **General economy of design**

a. The expressive power of the formalism is part of the analysis: Wherever possible, the system should not allow possibilities that are linguistically unattested.[13]
b. The system should be non-redundant: the same information should not be encoded in two places at the same time.

3.2 Theta clusters as event modifiers

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's 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, *run*(e) is true if e is an event of running.

Thematic roles are represented as relations between this event and the appropriate participant; for example, the relation $\text{Ag}(e, \text{John}')$ 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. Sentence (27a) is interpreted as in (b).

(27) a. John kicked the ball.

[12] Note in particular that theta roles are therefore associated with V^0, not with the NP that satisfies them. They can be written on the corresponding NP only as a matter of informal practice, indicating the use that will be made of them in the course of the derivation.

This is not the only way to approach the semantics of thematic relationships. While Landman's (2000) event semantics is purely local, in the present sense, 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 addresses this issue while obeying locality considerations.

[13] This is the common strategy in generative linguistics: "Restricted" formal devices are preferable to more powerful ones, with the goal of accounting for attested patterns but ruling out unattested ones as underivable.

In particular, as discussed in section 2.2, the ordering of thematic arguments is predictable according to general rules (Reinhart's CS merging instructions). Allowing argument order to be built into the lexical entry of each verb would not only be redundant: it would raise the possibility of verbs whose arguments appear in unattested orders, e.g., Patient above Agent.

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

This says that e is an event of kicking, whose agent and patient were John and the ball b, respectively.

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; we can write [+m+c](e, J) instead of *Agent*(e, J), etc., yielding formula (28) in place of (27b).

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

We can even continue to use *Agent* as an informal synonym for [+m+c], and write (27b) but read it as (28).[14]

Parsons's presentation of event semantics is informal. While I will use as a starting point the formalization developed by Landman (2000), there exist numerous other formalizations along similar lines (cf. Link 1998:ch. 10,11, for example). The proposal given in section 3.4 should be compatible with any of them.

3.3 Events and concepts

An important conceptual issue arises with respect to theta grid operations that apply in the lexicon. In Reinhart's terminology, these derive new "concepts", e.g., reflexivization of the verb *love* may give us the self-love concept. But if these new concepts are produced by the application of general rules, they should be definable in terms of the base concept: they can also be viewed as derived predication structures for the same base concept. What should be the relationship between the concepts represented by the base and the lexically derived verb, and how does it differ from their relationship to the output of syntactic-component arity operations?

While this is an important question, I believe that as stated above it is not clearly posed. The real issue it raises is what we mean by "concept." This is in essence a philopsophical question that I do not wish to take a position on, and so I will also not take a position on how the concept represented by the output of lexicon operations is related to the base concept. The question we must address here is: do the concepts associated with lexically derived verbs also require a new *event type*? For example, consider an event type predicate *pinch(e),* which applies truly to events of (transitive) pinching. Now consider an event of self-pinching, e_0. Should the same event type predicate apply to it, or do we need a new predicate, *self-pinch(e)*? From the point of view of modeling 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. The meaning of a reflexive verb *Refl(pinch)* is then given given in terms of the (transitive) *pinch(e)* event type, as follows:

(29) $[\text{REFL(pinch)}](e) = \lambda x\,\exists e\,\text{pinch}(e) \wedge \text{Ag}(e, x) \wedge \text{Pat}(e, x)$

I will adopt this strategy for our model: Lexicon operations, whether or not they can be said to derive new concepts, do not change the underlying event type predicate.

Note, however, that 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 lexicon; their meaning can in principle involve any event type, regardless of the meaning of the base verb.[15] Derived forms with compositional meaning are not listed; the definition of the lexical operation

[14]This approach is insufficient if we make a distinction between (semantic) theta roles and feature clusters as discussed in section 2.1, in other words, if a single feature cluster is associated with multiple theta role predicates. Accounting for this complication is technically straightforward to arrange, by allowing for a richer inventory of semantic theta roles which are *projected* to the feature clusters of the theta system, although it complicates the implementation of the theta system's functions. As already noted, I will make the simpler assumption that the feature clusters *are* the theta roles.

[15]I assume here a notion of "listedness" as proposed by Di Sciullo and Williams (1987). Not only roots but objects of arbitrary complexity, including phrases (idioms), may be listed in the lexicon. The listed objects are those whose meaning is not compositionally predictable.

must specify a way of computing the meaning of the derived verb as a function of the meaning of the base verb.

3.4 An event semantics for verbs

The semantics of the Theta System do not require an elaborate theory of event semantics. It is enough to define eventuality variables as simple objects of which various relations may be predicated. The following simplified description, which incorporates the essential ingredients of Landman's (2000) event semantics, is sufficient for our purposes.[16]

(30) **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.[17]

b. There is a set V of *event type* predicates over the elements of E. These are unary predicates corresponding to the core meaning of verbs. For example, the event type predicate $\lambda x\, \mathit{run}(x)$ is true of all and only those elements of E that are events of running. Event type predicates are of type <s, t>.[18]

c. There is a finite set R of *roles* (*Agent, Patient,* etc.), which are relations between the sets E and A. Roles are predicates of type <e, st>.[19] I will assume that these roles correspond to the feature clusters of theta theory.

d. **Role specification:** Lexical constraints (general and/or lexeme-specific) constrain which roles are compatible with a lexical item and which are obligatory with it.

e. The eventuality variable is eventually bound by existential closure. Landman leaves the details open, allowing existential closure to apply at the VP or the IP level, depending on one's syntactic analysis.[20]

Examples of an event type and a theta role predicate are given in (31a) and (b). The grammar translates each verb into a formula that conjoins its event type and theta role predicates (in the appropriate order), as shown in examples (c) and (d).

(31) a. $\lambda e\, \mathrm{break}(e)$.

b. $\lambda x_e\, \lambda e_s\, \mathrm{Ag}(e, x)$.

c. *walk* $\rightarrow \lambda x\, \lambda e\, (\mathrm{walk}(e) \wedge \mathrm{Ag}(e, x))$

d. *kiss* $\rightarrow \lambda y\, \lambda x\, \lambda e\, (\mathrm{kiss}(e) \wedge \mathrm{Ag}(e, x) \wedge \mathrm{Th}(e, y))$

The above description should give the reader an idea of the formal properties we expect from our event semantics. Our concern is now with deriving the representations of verbs shown in (c) and (d)

[16] 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, rather than distributing them to its syntactic arguments.

While Landman's goal was to deal with the properties of plurals, these are not relevant for our present purposes and are mostly suppressed 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 here given different names, 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.

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

[18] Event types as written here are characteristic functions on sets of events. Landman actually represents them as sets of events. The two formulations diverge only until we apply existential closure over the eventuality variable.

[19] Landman follows the common alternative of representing thematic roles as (partial) functions from events to individuals, e.g., $\mathrm{Ag}(e) = \mathrm{John}'$. The two notations are equivalent if we require that if a thematic role is instantiated for a particular event, then it is instantiated by exactly one (possibly plural) individual. This is Landman's (2000:38) "unique role requirement".

[20] Landman requires existential closure to apply after in situ arguments (those that have not undergone QR) have been combined with the verb, but before combining with quantificational arguments. The timing of event closure also has implications for interpretation, but these considerations are beyond the scope of our present concerns.

above, and with allowing the machinery of the theta system to control the "role specification" functions that Landman assumes. Clearly it is not enough to simply list the forms in (c) and (d) in the lexical entry of each verb: Such a representation involves a particular set of thematic arguments, projected in a fixed order. It is not sufficient for the needs of the Theta System, which provides an account of *alternations* in the argument structure of verbs, based on a single underlying lexical entry. Our goal, then, is to show how the above translations can be derived from suitable stored lexical entries, in a way that allows us to account for the regularities in verbal argument structure that the Theta System identifies.

Our model must reflect the following properties of the Theta System:

(32) a. The theta roles listed in a verb's lexical entry may be added to, deleted, or modified in various ways;
b. subject to certain conditions, some roles may remain unexpressed;
c. various syntactic properties (unaccusativity, Case assignment) are determined on the basis of the theta roles *and* the modifications they have undergone;
d. finally, the order of argument projection must be determined by rule.

The last property merits further comment. According to the Theta System, the order of thematic arguments is determined by the *CS merging instructions,* which apply after lexical marking and the arity operations. It follows that this order is not hard-wired into the lexical entry of the verb; otherwise, it would be meaningless to "calculate" it again. On the other hand, the semantic form which we want to define is a function, encoded as a lambda term; and the arguments of lambda formulas appear in a fixed order. From an implementation perspective, this means that the ultimate semantic form (as given in (31)) cannot be the representation we start with: there must be a *change of representation,* at some point between the original lexical entry and the point at which semantic interpretation occurs. In the next section I consider some linguistic motivations in support of this conclusion.

3.5 The change of representation

We begin with the question of how a verb's thematic grid is stored in the mental dictionary (the verb's *lexical entry*). It is well-known that thematic roles are not mapped idiosyncratically to verbal argument positions; their arrangement is subject to some sort of universal constraints. For example, no verb may assign the Agent role to its object and the Patient role to its subject. The Theta System adopts a version of the Universal Alignment Hypothesis (UAH), maintaining that the ordering and projection (internal or external) of the theta roles of any given verb is subject to universal constraints.[21] In the Theta System this is accomplished by means of a general procedure, the CS merging instructions given in (6), which (partially) determines the syntactic projection of any verb's arguments.

To allow this ordering procedure to apply, we must assume that the order of arguments is not specified in the lexical entry of the verb; otherwise, it would be meaningless to "calculate" it again. We begin with a lexical entry whose arguments have no intrinsic order; the argument projection rules eventually build an ordered representation,possibly after the application of lexical operations that involve the addition, deletion or manipulation of some clusters of the original lexical entry.[22] 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 (see, for example, Di Sciullo and Williams 1987:where?), 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

[21] Unlike some stronger claims, the UAH does not assume an invariant structural position for each theta role.

[22] Constraints on ordering are not incompatible with storing argument order in the verb's lexical entry: they would then take the form of well-formedness constraints on lexical entries, i.e., of a filter. In other words, lexical entries would hard-wire the order of their arguments, but only orders consistent with the CS-merging rules would be permitted. Such a "declarative" approach, however, would be at odds with the Theta System's strongly procedural flavor. The CS merging instructions apply after lexical 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.

exceptions to exist, such as verbs whose Agent and Patient occur in reverse order. But such verbs are unattested. We conclude:

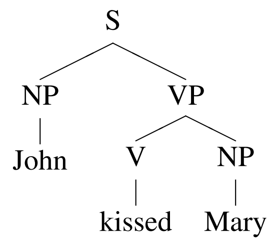
(33) The order of thematic arguments of a verb is not specified in its lexical entry.

Recall that our objective is to provide an explicit account of the semantics of argument structure operations, from the lexicon to the final logical form. The end product of the account will be a truth-conditional interpretation, expressible in terms of lambda formulas. But the arguments of lambda formulas are inherently ordered: The semantic form for *kiss,* given in (31) and repeated below, specifies that the closest (internal) argument of *kiss* is its Theme, and the higher argument is its Agent. This is more information than the lexical entry should encode, since it is just as easy to define verbs whose Agent and Theme arguments occur in the reverse order.

(34) *kiss* $\rightarrow \lambda y\, \lambda x\, \lambda e\, (\text{kiss}(e) \wedge \text{Ag}(e, x) \wedge \text{Th}(e, y))$

The fixed order of arguments is not an accident of the implementation; it is inherent in syntactic and semantic phenomena, as well as in the theoretical frameworks employed in their analysis. In the grammar architecture assumed by the Theta System, lexical objects are collected into the Numeration, and assembled into a structured syntactic object through a process of syntactic derivation. The semantics of the constituents and the resulting aggregate object correspond to objects in some intensional logic, which we can write as formulas of lambda calculus. We leave open the question of whether there is a mapping from syntax to semantics at the end of syntactic derivation, or whether semantic representations are built up in parallel with the syntax. In such syntactic and semantic representations, verbal arguments are inherently ordered. In syntax, this order is encoded in a hierarchical (structural) way. In a given syntactic tree, each argument appears in just one structural position, which determines the order in which it forms constituents with the verb and the other arguments. (A tree, of course, can be transformed into another tree through movement). The formulas of Montague grammar similarly fix the order in which the arguments of the verb should be saturated. For example, the tree in (35) specifies the syntactic projection of the arguments of the verb *kissed.* Its constituent structure determines the order in which the semantic form of its constituents will combine. Arguments are associated with the proper thematic role because the semantic form of each node indicates the order in which its arguments should be supplied; for example, the semantic form of the verb *kiss,* given above, specifies that it must be combined with the Theme *Mary* before the Agent *John.*

(35)



In short, argument order is not specified in the lexical entry of verbs; but by the time a semantic interepretation for the sentence is constructed, we are dealing with a form that imposes an explicit hierarchy (order) on the verb's arguments. It follows that the verb's stored lexical entry is *not* in this final form. We are led to the conclusion that there is a *change in representation,* somewhere between the basic lexical entry and the logical form of the sentence. For convenience, I will refer to this change in representation as *translation.*

3.6 The process

We have arrived at the following scenario: At the lexical entry, the theta grid of the verb is representable as 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 original representation is translated into a semantic form, expressible as a lambda term, which gives the proper event semantics for the verb (or whatever larger unit is translated).

Along with the inventory of theta roles, each representation 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 would, among other things, include formal features as required by the Minimalist program. But each form must have some information about the verb's semantics and about its arguments. It is this information we focus on, with the understanding that it is just part of a larger structure.

The relevant component of the lexical entry for *kick* contains the event type predicate that identifies an event variable as an event of kicking, and a set of theta roles that will eventually be translated into the conjuncts of formula (34).

(36) **Lexical entry for *kiss***
Event type: $\lambda s\, \mathrm{kiss}(s)$
Roles: { [+c+m], [–c–m] }

Given the general constraint that Agents must project higher than Themes, it can be seen that the above contains enough information to construct formula (34). Each theta role represents a relation between individuals and events, as described in section 3.2; the resulting propositions are conjoined in the appropriate order to give (34).

After translation, theta structure information might be present twice, in the original theta grid and in the ordered representation. To ensure non-redundancy and consistency, I will assume that the ordered form replaces the original, unordered representation. The relevant content of the lexical entry is consumed ("checked") during the translation step, and is no longer accessible directly. Any subsequent morphosyntactic or semantic operations must work by manipulating the new representation. In particular, any arity operations applying after translation must alter argument structure by manipulating logical formulas; their semantic effect should be expressible as operators on formulas. To summarize,

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

Although a lambda formula contains ordering information, it does not contain any syntactic information about the projection of its arguments, beyond their order: The formulas for an unaccusative and an unergative verb have combinatorially indistinguishable formulas, and two-argument unaccusatives are similarly indistinguishable from ordinary transitives. I will assume that when the set of theta roles is assembled into a lambda formula, the relevant projection information is computed and encoded in the form of syntactic features.[23]

4 A model for building verb denotations from lexical entries

We now sketch a simple model that can carry out the computations required by the Theta System, and derives the appropriate semantic forms for verbs. The process of lexical derivation will proceed as shown in (38). Its components are described in more detail below.

(38) **The derivation process**

1. Begin with a copy of a verb's stored lexical entry.
2. Add the ACC feature if appropriate.
3. Apply any desired argument structure operations.
4. Immediately before the translation step, decide whether to project an external argument.

[23]We return to this issue in section 4.4.

5. Perform the translation step, translating each role into a theta-role predicate and conjoining with the event type in the appropriate order.
6. Any further operations will apply to the translated semantic form, plus remaining untranslated information such as the ACC feature.
7. Apply existential closure to the event variable.

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).

(39) a. An **event type** predicate, e.g., $\lambda e\, open(e)$.
b. A *set* T, whose elements represent the theta roles of the verb.

During the course of lexical derivation, an object representing a copy of the stored lexical entry is modified in various ways. Let us refer to this object (initially a copy of the stored lexical entry) as a Working Copy, or *Working Structure, WS.* In addition to the information inherited from the lexical entry, the WS can store values for the features ACC and EXT, which are assigned by the lexicon marking rules. The **arity operations** manipulate the WS's copy of T, deleting elements or inserting others representing the result of the operation, and may remove the ACC feature. 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.

The possible elements of T are the following:

(40) a. The eight clusters of theta theory.
b. For each cluster α, elements *Sat(α)* and *Arb(α).*
c. For each (unordered) pair of clusters α and β, a "bundled" element $\alpha\beta$.
d. A special element *Del.*

We assume that only the eight elementary clusters occur in stored lexical entries. The modified elements *Sat(α),* etc., serve to preserve a history of the lexical derivation of the verb. They are eventually translated into the appropriate logical formulas, as described below.

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 presented 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. Rather than model the indices and the rather complicated ordering constraints that they involve, we will achieve their effect by direct reference to the values of the theta clusters.

4.2 The ACC feature

The ACC feature determines whether a verb will assign accusative case. It is assigned to the verb's WS at the beginning of the derivation, but may be removed by subsequent arity operations. The Lexicon Marking Rule (5c) specifies that ACC is added if the verb (a) has two or more arguments, and (b) its arguments include both an all-plus cluster and a fully specified cluster with the [/−c] feature (thus, either [−c+m] or [−c−m]).

4.3 Arity operations

Argument structure operations can be language-specific, and differ in their details. But they make use of a universal set of "arity operations." Because of the change in representation involved in the translation step, pre- and post- translation arity operators will manipulate different objects, and must be defined differently. While a system in which arity operations only apply in one domain would be conceptually simpler, the empirical facts indicate the need for two domains of application. The theta system posits two domains of application ("lexical" and "syntactic"), and I will show below that these should be identified with pre- and post-translation application.

Early ("lexical") arity operations manipulate the WS, modifying its contents as follows.

(41) **The lexical arity operations.**

a. **Saturation**

(i) Replace an element α with *Sat*(α).

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

b. **Arbitrarization**

(i) Replace an element α with *Arb*(α).

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

c. **Expletivization (External Reduction)**

(i) Replace a [+c] element with *Del*.[24]

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

d. **Reflexivization**

(i) Replace two elements α, β with the element $\alpha\beta$.

(ii) Remove the ACC feature, 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.[25]

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

(iii) Add an ACC feature, if appropriate.

f. **Non-realization**

Under the conditions spelled out by Reinhart (2002), replace a cluster α optionally or obligatorily with *Del*.[26]

While the theta system defines Reduction as an operation that completely eliminates a cluster from a verb's theta grid, it is not so implemented above. Instead, a dummy element is left behind to indicate that a deletion has occurred. In this way T maintains a partial record of the derivational history of the verb, which greatly simplifies our determination of internal vs. external projection (next section).

4.4 Internal and external projection

The Theta System's rules for determining internal and external projection are subject to elaborate rule ordering constraints: The indexing rules, 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 indexing rules. But the indexing rules are not affected to Reduction and Saturation, which remove an argument; a verb left with just one argument due to Reduction will nevertheless be subject to the indexing rules.

Rather than adopt the necessary rule ordering and keep track of all the required auxiliary information, we follow here an insight by Siloni (2002), who points out that we can account for the varying impact of the arity operations if we assume that the indexing rules apply to the "maximal" theta-grid of the verb: Causativization, which extends the theta grid, is visible, while Reduction, which reduces it, is not. But this criterion cannot be applied in a sequential model that completely eliminates a cluster: at the time the indexing rules are applied, it is necessary to know whether a cluster was present but has been deleted.

[24] This solution is chosen over immediate deletion in order to eliminate the need for certain rule ordering restrictions, discussed in section 4.4.

[25] If we allow multiple theta roles for some clusters, we need a corresponding mapping between theta roles (or a default role for each cluster, which serves the same purpose).

[26] Non-realization does not correspond to an explicit rule of the Theta System. Reinhart holds that certain co-occurrence conditions simply render the realization of certain clusters optional, or in some cases impossible.

The model of "deletion" that we have adopted, which replaces deleted clusters with the special element *Del,* maintains a history of the derivation and makes the following simple algorithm possible. Note that the indices described below are used as soon as they are computed; therefore they do not need to be stored in the WS.

(42) **Merging instructions** (without stored indices)
a. If there is only one element in T, it will merge externally.
b. Otherwise, compute indices as follows, and use the *CS merging instructions* (6) to determine whether an argument will merge externally.

(43) **Lexicon marking rules** for elements of T.
a. Elementary clusters are indexed according to the *lexicon marking rules* (5a,b).
b. Elements of the form *Sat*(α) are considered to have the same index as α would.
c. The bundled elements are considered to have index 1 if and only if one of the bundled clusters has index 1.
d. The element *Del* is ignored.

Our semantic implementation does not provide a way of distinguishing internal and external arguments. A lambda form contains ordering information, but does not determine the syntactic position of its arguments, beyond their order: An unaccusative and an unergative verb have combinatorially indistinguishable formulas, and two-argument unaccusatives are similarly indistinguishable from ordinary transitives. I will assume that this information is a syntactic property, and is encoded by means of an additional feature passed to the syntax after translation.[27]

4.5 The translation step

The translation 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, Proj_n, which add a theta role predicate to a partially constructed verb denotation.[28] The elementary one, Proj_0, adds a theta role to an event type predicate. Higher-order projection operators operate on denotations that already include some arguments.

(44) **Argument projection operators**
a. $\mathit{Proj}_0 = \lambda R_{e,st}\, \lambda P_{st}\, [\lambda x_e\, \lambda e_s\, P(e) \wedge R(x, e)]$.
b. $\mathit{Proj}_1 = \lambda R_{e,st}\, \lambda P_{e,st}\, [\lambda z_e\, \mathit{Proj}_0(R, P(z))]$.
c. ...
d. $\mathit{Proj}_n = \lambda R_{e,st}\, \lambda P_{e,\ldots st}\, [\lambda z_e\, \mathit{Proj}_{n-1}(R, P(z))]$.

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. (Proj_2 is the highest needed for ditransitive verbs).

We can then carry out the translation procedure as follows:

(45) **Translation procedure**

[27] Chierchia (1989) 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.

Rather than adopt Chierchia's system with its attendant complications, I will assume that projection information is transferred to the syntactic component at translation time as a feature.

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.

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

1. Begin with the verb's event type predicate.
2. If T contains an element *Del,* remove it.
3. Select the verb's theta clusters, one by one, in an order consistent with the CS merging instructions (lowest argument first).
4. Use the appropriate projection operator to combine the theta cluster with the partial verb denotation, as follows:
   (a) Elementary clusters are merged as the corresponding theta role predicate.
   (b) Bundled clusters are merged as a conjunction of predicates. For example, for the bundle [+c+m] $\oplus$ [−c−m] we merge the complex role predicate $\lambda x\, \lambda e\, (\text{Ag}(e, x) \wedge \text{Pat}(e, x))$.
   (c) For elements of the form *Sat(*α*)* and *Arb(*α*),* first merge the corresponding cluster α; then apply the corresponding existential closure operation.
5. Remove each cluster from T as it is used.

The result of the translation procedure is a function, expressed as a lambda form, with a determinate hierarchical order for its arguments. It continues to be supplemented by features controlling, inter alia, accusative Case assignment and whether to project an external argument. While the system still needs to ensure that argument NPs are syntactically merged in the appropriate order, the semantics of the verb reflects the *output* of arity operations, and allows any linguistically licit argument order to be realized.

4.6 Examples

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

(46) a. **Lexical entry for *open***
Event type: $\lambda e\, \text{open}(e)$
Arguments: $T = \{[+\text{c}+\text{m}], [-\text{c}-\text{m}]\}$ (Agent, Theme)

b. **Derivation**
1. From lexical entry: $T = \{[+\text{c}+\text{m}], [-\text{c}-\text{m}]\}$
2. Add Acc feature: $T = \{[+\text{c}+\text{m}], [-\text{c}-\text{m}]\}$; Acc
3. Apply Reduction: $T = \{\text{Del}, [-\text{c}-\text{m}]\}$
4. External argument: T has 2 elements, so use merging instructions: project the surviving all-minus cluster internally
5. Translation: *open* $+\ \lambda x\, \lambda e\, \text{Theme}(e, x)$ =
$\textit{Proj}_0(\lambda x\, \lambda e\, \text{Theme}(e, x), \lambda e\, \text{open}(e)) =$
$\lambda x\, \lambda e\, \text{open}(e) \wedge \text{Theme}(e, x)$

For a more complex example of the projection operation, we illustrate the addition of the final (highest) argument of the three-argument verb *peel,* whose lexical entry is given in (47a).

(47) a. **Lexical entry for *peel:***
Event type: $\lambda e\, \text{peel}(e)$
Arguments: {[+c+m], [−c−m], [+c−m]} (= Agent, Pat, Instr)

b. **Derivation**
No arity operations; we suppress the details of how the first two arguments are assembled.

c. **Partially built denotation:**
$\lambda z\, \lambda y\, \lambda e\, (\text{peel}(e) \wedge \text{Instr}(e, z) \wedge \text{Pat}(e, y))$.

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

e. **Argument projection:**
1. peel(Pat, Instr) + [+c+m] =
2. $\textit{Proj}_2([+\text{c}+\text{m}], \text{peel}(\text{Pat}, \text{Instr})) =$
3. $\textit{Proj}_2(\lambda x\, \lambda e\, \text{Agent}(e, x), \lambda z\, \lambda y\, \lambda e\, (\text{peel}(e) \wedge \text{Instr}(e, z) \wedge \text{Pat}(e, y))) =$
4. $\lambda z'\, [\textit{Proj}_1(\lambda x\, \lambda e\, \text{Agent}(e, x), \lambda y\, \lambda e\, (\text{peel}(e) \wedge \text{Instr}(e, z') \wedge \text{Pat}(e, y)))] =$

5. $\lambda z' \, \lambda y' \, [Proj_0(\lambda x \, \lambda e \, \text{Agent}(e, x), \lambda e \, (\text{peel}(e) \wedge \text{Instr}(e, z') \wedge \text{Pat}(e, y')))] =$
6. $\lambda z' \, \lambda y' \, \lambda x \, \lambda e \, (\text{peel}(e) \wedge \text{Instr}(e, z') \wedge \text{Pat}(e, y') \wedge \text{Agent}(e, x))$.

5 Arity operations in the lexicon and in the syntax

5.1 When does translation happen?

We now consider the following question: at what point is the information in the lexical entry assembled into a lambda formula? Until now I have been vague about the exact point in which the translation step applies, assuming only that it must happen at some point between the lexical entry and the logical form. We now consider the following possibilities:

1. **Early translation:** The translation is made at the earlest possible moment, before any arity operations on the verb entry.
2. **Translation at lexical insertion:** The translation is made just prior to insertion of the verb in the numeration; that is, after arity operations that apply in the lexicon but before operations that apply in the syntax.
3. **Late translation:** The translation is made at some late point during syntactic derivation, after all syntactic arity operations.

Early translation would mean that a lambda form is built directly from the verb's lexical entry immediately; all arity operations must work by manipulating the resulting lambda formula. Obviously this would render useless the arity operations defined in section 4.3, which operate on a modified copy of the lexical entry. Late translation would most simply mean translation of the entire sentence at LF, at the end of syntactic derivation; but it could also be translation at some intermediate point.

The timing of the translation step determines the form of semantic information at each stage of lexical-syntactic derivation: before translation, arity operations (and other operations that manipulate the lexical item) must operate on the structured representation we have defined; after translation, they must manipulate the semantic objects (lambda forms) built by the translation. 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 condition" (37) 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:

(48) **Post-translation condition:** After translation is complete, all arity operations must operate on the resulting formula; their semantic effect must be expressible as operators on suitable formulas of some intensional logic.

The early translation option has the advantage that arity operations manipulate the same sort of objects (lambda formulas) 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 syntactic component counterpart. The late translation option also has this advantage, since both types of rules manipulate structured lexical entries rather than formulas. On the other hand, translation at lexical insertion offers the potential of accounting for the differences between lexical and syntactic operations as consequences the fact that they apply to different kinds of objects. We will see that exploiting this option leads to a surprising number of correct predictions.[29]

[29] A fourth alternative, proposed by Tany Reinhart (personal communication), involves translation at the VP level. Because syntactic arity operations involve the external argument, this option is combinatorially similar to translation at lexical insertion. Its potential advantages and disadvantages involve factors that go beyond simple role projection, and will not be considered here.

A single lambda formula is just a function, and is harder to manipulate as a formal object than a set of distinct theta clusters and other features. The following are natural consequences of the properties of lambda formulas; it is possible to design a system that will circumvent them, but it would involve ad hoc work-arounds and would be considerably more complex than the system outlined in section 4.

(49) **Limitations of operations on formulas**
  a. A theta role cannot be completely deleted from a formula, nor replaced with another one.
  b. No operation on a formula can be restricted to particular thematic types.[30]

(50) **Operations that can be readily applied** to the arguments of logical formulas
  a. Existential closure (saturation).
  b. Existential closure by a variable of a special, "arbitrary" type (arbitrarization).
  c. Reordering (apparently not attested).
  d. Identification with another argument (reflexivization).
  e. Insertion of additional arguments (expansion).

I will not prove these limitations; I expect that their correctness is obvious to readers familiar with the lambda calculus. They are all consequences of the fact that operations on lambda formulas manipulate functions, not strings. The truth-valued function $\lambda x\,\exists e\,\text{run}(e) \wedge \text{Ag}(e,x)$, extensionally considered, is simply the function that returns *True* for all individuals that are Agents of some event of running. This function could be written in other ways (for example, as a large truth-value table), and the set it picks out might also be the characteristic set of other functions, which do not involve *Agent* in their definition. Consequently it is not possible to write a formula that recognizes all and only lambda forms containing *Agent* as one of their conjuncts: the corresponding function simply does not contain enough information.

All of the above operations are possible with a suitably constructed system that represents arguments as a set, such as the system defined in section 4. In short, operations on structured lexical entries are potentially more powerful than operations on lambda forms. This provides us with a way to decide where the translation step could be located: It cannot be earlier than any arity operation that requires access to the structured lexical entry.

5.2 The distribution of arity operations

When we consider the arity operations in the theta system's inventory, we find that there are indeed arity operations that cannot be written as operations on lambda forms: Reduction (including Expletivization), which completely removes a theta cluster from the verb's grid, and Causativization, which replaces a cluster with its "decausativized" 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.

1. **Expletivization**

   Expletivization is reduction of a [+c] theta cluster (necessarily external). The cluster is completely removed from the theta grid of the verb. (Reinhart 2002). This is the operation that derives unaccusatives. Our system predicts that Expletivization could not be a syntactic operation for two reasons: it removes a theta role (which would require deletion of the corresponding conjunct from the assembled lambda formula), and it is sensitive to the presence of a specific cluster.

   As predicted, this operation only takes place in the lexicon. It is easily formulated as an operation on the WS: the element [+c] is simply removed from the set of roles (and replaced by *Del*).

[30] This is a rather drastic limitation, and I am not sure if it can be completely maintained. For example, it has been claimed that syntactic reflexivization requires an animate subject. It should be stressed that this limitation applies to formulas in themselves. An operation can still be restricted to verbs that project an external argument, for example, by consulting projection information that has been encoded in the syntax.

It follows that the limitation itself could be circumvented by maintaining the required information in the form of "features"; but the simplest design, which follows the non-redundancy principle 37 and does not redundantly encode theta role features after translation, is to be preferred if it can be maintained.

2. **Causativization**

   Causativization mutates one theta cluster into another: It replaces a cluster containing a +c feature, if there is one, with the corresponding cluster containing −c (so that [+c+m] becomes [−c+m], etc.). This is not a licit syntactic operation, and correspondingly Causativization is restricted to the lexicon. Causativization also adds a new [+c+m], i.e., Agent role. (Addition of an Agent is easily expressed as an operation on lambda forms).[31]

3. **Saturation and arbitrarization**

   These operations existentially close an argument, and can be readily applied in the syntax. The passivization operation in the theta system is an example. It is easy to write a family of operators that saturate the arguments of predicates of different arities. For example, the operator defined in (51a) will saturate the external argument of a transitive verb (before it has combined with any arguments).

   (51) a. $\text{SAT}_e = \lambda P_{<e,<e,st>>} [\lambda x\, \lambda e\, \exists y\, P(x)(y)(e)]$ (Type: <<e, <e, st>>, <e, st>>)

   b. $\text{SAT}_e(\lambda x\, \lambda y\, \lambda e\, \text{break}(e) \wedge \text{Ag}(e, y) \wedge \text{Pat}(e, x)) = \lambda x\, \lambda e\, \exists y\, \text{break}(e) \wedge \text{Ag}(e, y) \wedge \text{Pat}(e, x)$

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

4. **Reflexivization**

   Reflexivization applies in the lexicon as well as in the syntax. It is straightforward to write it as an operator on formulas, given Reinhart's recent conclusion that reflexivization involves identification of two roles (and not reduction of the internal role, as in earlier accounts).[32]

   An operator that reflexivizes a transitive verb can be written as follows:

   (53) REFL = $\lambda P_{<e,<e,st>>} [\lambda x\, \lambda e\, P(x)(x)(e)]$ (Type: <<e, <e, st>>, <e, st>>)

Explaining the distribution We conclude that the translation step 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 translation in fact takes place at the dividing line between lexical and syntactic operations, just prior to lexical insertion and the beginning of morphosyntactic derivation. We then have a way of correctly predicting which arity operations are restricted to the lexicon and which may occur in the syntax: Reduction and Causativization are restricted to the lexicon *because* their operation cannot be expressed as an operator on functions (lambda formulas). The remaining arity operations, which are not so restricted, apply in the syntax as well as the lexicon.

Let us reiterate here that a suitably constructed system could provide a work-around for these limitations; but the present system, which involves nothing more than the elements necessary to assemble ordinary event semantics representations starting from an unordered set of theta clusters that may be subject to arity operations, correctly predicts the distribution of arity operations.

[31]Reinhart notes that there is an unrelated, syntactic "causativization" construction; it has very different properties from the arity operation discussed here.

[32]Reinhart sees 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 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 would seem to disappear 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.

5.3 An alternative explanation

In the previous section I showed that an arity operation occurs in the syntax just if the manipulations it performs can be expressed as an operation on lambda forms; Reduction and Causativization 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 limited *because* syntactic arity operations must be expressible as operators on lambda forms.

Siloni (2002) proposes an alternative explanation of this 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."

(54) **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.

Both explanations rule out deletions or modifications of feature clusters in the syntax; the account of the previous section 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. If the two accounts made identical predictions, choosing between them would 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, as in other languages, "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). Some Kichaga examples are shown in (55).

(55) Applicative: Kichaga (Bresnan and Moshi 1990)
a. N-ấ-ĩ́-ly-à k-élyà.
Foc-1sg-Pres-eat-FV 7-food
'He/She is eating food'
b. N-ấ-ĩ́-lyì-í-à m̀-kà k-élyà.
Foc-1sg-Pres-eat-Appl-FV 1-wife 7-food
'He/She is eating food on (to the benefit/detriment of) his wife'
c. N-ấ-ĩ́-lyì-í-à mà-wòkő k-êlyâ.
Foc-1sg-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 morphosyntactic operation, as seems quite plausible, then it contradicts the proposed Lexicon Interface Guideline (54).

The system proposed in section 4 allows applicative formation to be formulated as a syntactic-component operation, and therefore predicts that it should be possible. 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 defined in (44) do. While the argument inserted by the projection operators is always higher (merges later) than any existing arguments, we can also write a formula that adds an argument that merges first (or, indeed, in any intermediate position). The following formula will add a Beneficiary role to a transitive verb:

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

6 Conclusions

I have outlined a concrete proposal for embedding the Theta System in a simple account of event semantics. By posing 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 lexical and in the syntactic component.

It must be remembered that the restrictions I have associated with syntactic-component operations are suggested by the formal framework (Montague-style truth-conditional semantics), not by conceptual necessity. This account of the distribution of arity operations relies on the premise that the logical formulas employed are not just a convenient notation, but have expressive power that corresponds to the real formal power of the operations that take place during semantic composition. It can be argued that in finding a correlation between properties of the formal framework and actual properties of syntactic argument structure operations, we have obtained empirical support for our choice of formal framework: The correlation can be taken as evidence that the semantic representation of language does in fact have properties analogous to those of the formal framework.

7 Reminder

This ms. is still a draft. Your comments, corrections, and suggestions for improvement are especially welcome.

References

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

Chierchia, Gennaro. 1989. A semantics for unaccusatives and its syntactic consequences. Ms., Cornell University.

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.

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. 2002. Discontinuous reciprocals and symmetric events. Paper presented at the Anaphora Typology Workshop on Reciprocals, Utrecht University.

Dowty, David R. 1976. Montague grammar and the lexical decomposition of causative verbs. In *Montague grammar*, ed. Barbara H. Partee, 201–245. New York: Academic Press.

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

Kayne, Richard. 1975. *French syntax: The transformational cycle*. Cambridge, MA: MIT Press.

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.

Levin, Beth, and Malka Rappaport Hovav. 1995. *Unaccusativity: At the systax-lexical semantics interface*. Linguistic Inquiry Monograph 26. Cambridge, MA: MIT Press.

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

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.

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

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

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, and Tal Siloni. 2003. Thematic arity operations and parametric variations. Ms., Utrecht institute of Linguistics OTS and Tel-Aviv University, May 2003.

Reinhart, Tanya, and Tal Siloni. Forthcoming. Against the unaccusative analysis of reflexives. In *Studies on unaccusativity: The syntax-lexicon interface*. Cambridge University Press.

Siloni, Tal. 2001. Reciprocal verbs. In *Proceedings of the Israel Association of Theoretical Linguistics*, ed. Yehuda N. Falk. Online publication, available at *http://atar.mscc.huji.ac.il/~english/IATL/17/*.

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

Utrecht institute of Linguistics OTS
Trans 10
3512 JK Utrecht
The Netherlands
alexis.dimitriadis@let.uu.nl