Non-subcategorized arguments: a two component event-structural account*

Sergei Tatevosov
Lomonosov Moscow State University

1. Non-subcategorized arguments

A well-known illustration of the phenomenon I am interested in comes from English resultatives. In (1), *the teapot* cannot be a subcategorized argument of *drank*:

(1) a. We drank the teapot / #the juice dry. (Kratzer 2005)
  b. #We drank the teapot.
  c. We drank the juice.

*Drink* is a transitive verb. Its internal argument must denote a drinkable entity, so (1b) only makes sense on the odd interpretation where *the teapot* is a name for some kind of liquid. However, (1a) is a clause where *the teapot* is licensed, but the subcategorized argument of *drink*, *the juice* in (1c), is not.

In Russian and other Slavic languages, there is a class of prefixes, called resultative prefixes, that exhibit a pattern strikingly similar to what we observe with English resultatives. The verb stem *ed-* ‘eat’ in (2a) subcategorizes for a DP denoting eatable substances. With some resultative prefixes, its argument structure remains intact, (2b). But with others, just like what happens with Resultative XPs, an argument not subcategorized for by the verb is projected, (2c).

(2) a. Volodja el jabloko / #puzo
      V. eat.PST apple.ACC belly.ACC
      ‘Volodja was eating/ate an apple/ #a belly.’

  b. Volodja s’-el jabloko.
     V. PRF-eat.PST apple.ACC
     ‘Volodja ate an apple.’

*I am grateful to Pavel Grashchenkov, Sabine Iatridou, Ekaterina Lyutikova, Martin Walkow, the audience at NELS 45 and MGU Phonology and Morphology Workshop for comments and discussion. The study has been supported by Russian Foundation for Basic Research, grant #14-06-00435.
This similarity can be naturally attributed to the fact that the role of prefixes like na- and s- in (2b-c) is essentially similar as that of resultative XPs in English and other languages. As is extensively argued in the literature (Babko-Malaya 1999, Romanova, 2006, Svenonius 2004, Žaucer 2009), prefixes of this type introduce a result state description to the semantic representation. Therefore, DPs not subcategorized for by the verb stem can be thought of as licensed as arguments of a result state expression in both Russian and English (Levin, Rappaport Hovav 2001, Rothstein 2004, Kratzer 2005):

(3)  

a. \([ VP \text{ drink } [XP \text{ the teapot dry}]]\)  
b. \([ VP \text{ ed- ‘eat’ [XP puzo ‘a belly’ na-]}]\)

There is an observation, however, that seems to be problematic for the view instantiated in (3). *The teapot* in (1a) is not just a holder of the result state of being dry, as *the belly* in (2c) is not just a holder of the result state of being existent. The same two arguments are understood as undergoing change in the course of drinking and eating events, the change that culminates when the result state is attained. (1a), for example, involves a process of the teapot becoming empty (cf. the meaning of the same sentence in the progressive: *We are drinking the teapot empty*). Likewise, (2c) makes reference to the process of the belly being effected. In other words, (4) holds:

(4) An entity that acquires a result state (*the teapot* in (1a) and *a belly* in (2c)) is identical to an entity that undergoes change along a certain dimension in the course of the event.

It turns out, however, that (4) is not derivable under (3a-b) in any obvious way. We do not expect the teapot and the belly to be involved in the change of state subevent at all. A result state description only denotes a result state (e.g. ‘the teapot is dry’) and expresses no change of state on its own. On the other hand, verb stems only lexicalize changes that subcategorized arguments (e.g., *the juice* or *an apple*) undergo. If this were not the case, sentences like (1b) would be as appropriate as (1c), and we would not have any empirical reasons to identify some arguments as subcategorized in the first place. Therefore, if DPs like *the teapot* and *a belly* in (1a) and (2c) are only arguments of result state expressions, and nothing else is said, the resultative construction ends up having a meaning component that does not come from the meaning of its elements.

Below, I develop a proposal that accounts for this puzzle and offers a general way of dealing with non-subcategorized arguments. The discussion is based on Russian material but should be extendable to languages like English with minor technical adjustments.
2. Event structure

The key ingredient of the proposal is two-level architecture of event structure (ES): a complete ES is built up by putting together two components, lexical and structural. The former comes from the meaning of individual lexical items. The latter is created in the syntax. This puts the proposal in line with constructionist theories of ES, whereby the construction itself provides a structural template for an event description (Borer 2005, Zubizarreta, Oh 2007, Ramchand 2008, and others). I suggest, in the spirit of Ramchand 2008, that the subevental content of an event description comes as part of interpretation of the syntactic heads a vP is composed of. Specifically, v is associated with an activity subevent, V contributes a process/become subevent, and R introduces a result state. Denotations of v, V and R produce a structural, templatic meaning in the sense that subevents lack descriptive content. The latter appears when structural elements of ES are combined with lexical ones at the spell-out. Spell-out thus has both phonological and semantic consequences. I propose that it occurs in a cyclic fashion after every merge operation. Merge is interpreted by functional application, predicate modification and other common rules of construal. Spell-out is interpreted by the Match operation, which yields the intersection of the denotations of the two components:

\[
\text{Match}(\text{Struct, Lex}) = \| \text{Struct} \| \cap \| \text{Lex} \|
\]

This system is schematized in Figure 1, which represents a fragment of the structure I assign to prefixed configurations in Russian. At the first step of derivation, R is spelled out by the prefix, and the denotation of RP is formed by intersecting \| R \| and \| Prefix \| via Match. Then, V merges with RP to create V'. Semantically, \| V \| combines with the denotation of RP by functional application (FA). Then V is spelled out, and the lexical verb, \| V_{Lex} \|, matches with FA(\| V \|, \| RP \|), yielding the denotation of V'. Later on, the internal argument DP merges and gets spelled-out. At subsequent stages of derivation, not shown in (6), v merges with VP and projects, and the denotation of vP is computed in essentially the same way, through a series of FA and Match operations.

The system allows for a situation where a lexical item and a structural template do not have matching interpretations: \| \text{Struct} \| \cap \| \text{Lex} \| = \emptyset. If (and only if) this happens, the lexical element gets reinterpreted. It is this mechanism, also known as coercion, that plays a crucial role in accounting for the conditions where non-subcategorized arguments are licensed. I will return to this shortly.

---

1 The idea of two-level architecture is not new, see e.g. Rappaport Hovav and Levin 1998 and sources cited there. The current proposal inherits the distinction between lexical/ idiosyncratic and structural/ templatic aspects of meaning from the literature, but its theoretical implementation differs in many respects.

2 Given that there is one-to-one correspondence between sets and their characteristic functions, throughout this paper I switch back and forth between describing denotations in terms of sets and in terms of (Shonfinkeled) functions. For example, I will sloppily talk about "relations between individuals and events" and their intersections, but will represent them as functions of type \langle e, \langle v,t \rangle \rangle. The reader should have in mind that if, for example, both Struct and Lex are of type \langle e, \langle v,t \rangle \rangle, then \| \text{Struct} \| \cap \| \text{Lex} \| is to be understood as the function \lambda x. \lambda e. \| (x)(e) = \| \text{Lex} \| (x)(e) = 1.
3. Ingredients of ES

I follow Ramchand 2008 in assuming that transitive accomplishments contain the projections of R, V and v:

(7) \[
\ldots v \ldots \left[ \ldots V_{\text{ACC}} \ldots \left[ \ldots R \ldots \right] \right] \quad \text{ACCOMPLISHMENT EVENT TEMPLATE}
\]

The denotation of V assumed throughout this paper, however, differs from Ramchand’s. Part of it is the information that the internal argument undergoes change along certain dimension by a certain degree in the course of the event. I implement this by making use of the INCREASE relation between individuals, events and degrees proposed in Kennedy, Levin 2002. $V_{\text{ACC}}$ thus denotes a relation between a predicate of degrees, individuals and events in (8):

(8) \[
\| V_{\text{ACC}} \| = \lambda P <d,d> . \lambda x . \lambda e . \exists d [\text{process}(e) \land \text{INCREASE}(G(x))(d)(e) \land P(d)]
\]

where INCREASE$(G(x))(d)(e) = 1$ iff $G(x)(\text{FIN}(e)) - \text{’} G(x)(\text{INI}(e)) = d$.

In (8), G is a free variable over gradable properties (of type $<e, <i, d>>$) that gets its value from an assignment function.\(^3\) The value for the variable over degree predicates $P$ is provided by RP, which makes it obligatory in accomplishment structures for type reasons. Intuitively, V denotes a change, and thus needs to calculate what the endpoint of the change is. The role of RP is to contribute a key ingredient to this calculation. In that way, the

---

\(^3\)Analyzing G as a free variable seems to be justified for cases like (2b-c) in Russian, where the dimension of change is not specified by any overt material. This may be an oversimplification, given de-adjectival verbs like ‘lengthen’ or ‘empty’, where a gradable property is supplied by the adjectival stem. In a more extensive fragment, therefore, $\| V \|$ should be allowed to take G as an argument. For languages like English, likewise, this may be a necessary move at least for AP-resultatives like (1a), which contain an adjective specifying the scale of change. I will leave exploration of this line of the study for a future occasion, however.
presence of RP in an accomplishment event structure is derived rather than stipulated, unlike in Ramchand’s system.4

Turning to the denotation of R, I suggest that it contributes the set of maximal degrees, as shown in (9).

\[
\| R \| = \lambda d. \exists S[\text{max}(d)(S)], \text{ where } S \text{ is a scale with the maximal degree}
\]

(9) \[\| R \| = \lambda d. \exists S[\text{max}(d)(S)], \text{ where } S \text{ is a scale with the maximal degree}\]

I take lexical items (LIs) to be mappings between a phonological string and a set of grammatical and semantic features. The sample LIs are represented in (10).

\[
(10) \quad a. \quad \text{‘eat’: } / \text{ed-} / \leftrightarrow \{ \text{V, …, } \lambda x. \lambda e.eat(x)(e) \}
b. \quad \text{Prefix in (2b): } / \text{na-} / \leftrightarrow \{ \text{R, …, } \lambda d. d \in S_{\text{EFFECTED}} \}
c. \quad \text{Prefix in (2c): } / \text{s-} / \leftrightarrow \{ \text{R, …, } \lambda d. d \in S_{\text{CONSUMED}} \}
\]

One part of every lexical specification in (10) determines a category an LI can spell out (V in (10a), R in (10b-c)). Another part is a lexical meaning of an LI. In (10a), ‘eat’ denotes a (Davidsonian) relation between events and (consumed) objects. (I follow Kratzer (2003) in taking verbs to be names of unique relations between events and their internal arguments.) A resultative prefix contributes a set of degrees from a particular scale, as in (10b) and (10c). In that way the analysis captures the fact that prefixes co-vary with properties of scalar change. Specifically, whereas na- in (2c) is associated with a set of degrees from the effectedness scale, s- in (2b) brings in degrees from the scale of consumption.

Now we have everything we need to provide example derivations of sentences like (2a-c) and to account for the observed argument projection patterns.

4. Projection of a subcategorized argument

In (2b), as we saw, the internal argument is understood as subcategorized, since the same argument is available in combination with the prefixless stem in (2a). Assuming the overall architecture represented in Figure 1 and the meaning of LIs in (10a-c), we can easily derive exactly this result. First, the prefix in (10c) and \[\| R \|\] in (9) match according to the rule of construal in (5). The intersection of the set of maximal degrees from (9) and the set of degrees from the scale in (10c) is the singleton set containing the maximal degree of consumption:

\[
(11) \quad \text{Match}(\| R \|, \| s-\|) = \| [\text{RP } s-] \| = \lambda d. d \in S_{\text{CONSUMED}} \land \text{max}(d)(S_{\text{CONSUMED}})
\]

Depending on an assignment, G will be interpreted as a gradable property of being effected, consumed, affected, etc, as shown in (12). Intuitively, these properties

\[\text{A more elaborated version of the analysis is likely to require a further refinement. We may want RP to denote a set of states of an entity having a certain degree of a certain gradable property, rather than a set of degrees. This would be motivated by event structural considerations, e.g., the restitutive reading of ‘again’, whereby it only takes scope over a result state. The property of degrees as the denotation of RP will not serve a legitimate set of entities for ‘again’ to take scope over. For the current purposes, (7) will suffice, however.}\]
characterize “thematic classes of verbs”, e.g., creation verbs, consumption verbs, destruction verbs, verbs of combining and attaching and other classes of accomplishments (see, e.g., Levin 1993).  

\[(12)\]  
\[
\begin{align*}
&\text{a. } \| G \|^g_1 = \text{EFFECTED} \\
&\text{b. } \| G \|^g_2 = \text{CONSUMED} \\
&\text{c. } \| G \|^g_3 = \ldots
\end{align*}
\]

RP and V merge; their denotations from (11) and (8) are combined by FA.

\[(13)\]  
\[
| \text{Merge}(V, [RP \ s-]) |^g_2 = | V |^g_2 (| [RP \ s- ] |)^g_2 = \lambda x. \lambda e. \exists d[\text{process}(e) \land \text{INCREASE(CONSUMED}(x))(d)(e) \land d \in S_{\text{CONSUMED}} \land \max \(d)(S_{\text{CONSUMED}})]
\]

The denotation of \([ V [RP \ s-] ]\) is thus a relation between individuals and events where an individual is maximally consumed. According to (13), under \(g_2\) (and any assignment that is like \(g_2\) in this respect) the free variable over gradable properties is assigned CONSUMED, the property of being consumed, as a value. Under \(g_2\) we thus get a template for what is traditionally called consumption verbs. Note that under other assignments, e.g. \(g_1\), \([ V [RP \ s- ] ]\) will be an empty relation, since \(d\) will be construed as being a degree from two distinct scales at the same time. In this way, the distribution of a prefix is correctly predicted to co-vary with the dimension of (scalar) change.

Finally, (13) and the denotation of ‘eat’ from (10a) match:

\[(14)\]  
\[
\text{Match}(| ed- |^g_2, | [ V [RP \ s-] ] |^g_2) = \lambda x. \lambda e. \exists d[\text{eat}(x)(e) \land \text{process}(e) \land \text{INCREASE(CONSUMED}(x))(d)(e) \land d \in S_{\text{CONSUMED}} \land \max \(d)(S_{\text{CONSUMED}})]
\]

The outcome is the relation where an individual argument is an object of eating and of maximal consumption at the same time. This is exactly the meaning of ‘s’est’ in (2a). Saturating the external argument position, merging \(v\) and projecting an external argument in its specifier will complete the derivation of the denotation of \(vP\). For the sake of space, these straightforward steps are not shown here.

So far, we have seen how the proposed system deals with an easy case, the one where the internal argument is subcategorized for. The same result, however, is easily derivable within other, less complex systems. The two-level architecture gains certain empirical advantages in dealing with more complicated cases like (2c), where the argument does not appear to be subcategorized for by the verb stem. To this case I now turn.

5. **Projection of a “non-subcategorized” argument**

The first two steps of the derivation of (2c), ‘Volodja acquired a belly by eating’, are the same as before. The denotation of the prefix \(na-\) in (10b) and the denotation of \(R\) in (9) match, creating the singleton set containing the maximal degree on the effectedness scale.

---

5It should be pointed out that if gradable properties are made part of the semantics of ES, the notion of ‘verb class’, mostly used in an intuitive, pre-theoretical way in the literature, acquires a different weight. Within the current perspective, this notion is given more specific content: the classes are defined by descriptive characteristics of the scale of change that appears as part of the structural component of ES.
(15)  \[ [ V [RP na- ] ] = \lambda d. d \in S_{EFFECTED} \land \text{max}(d)(S_{EFFECTED}) \]

(15) merges with the denotation of V in (5). The [ V [RP na- ] ] constituent only denotes a non-empty relation if degrees in [ RP ] and [ V ] are construed as degrees on the same scale, which happens under all assignments \( g \) that are like \( g_1 \) from (12) as to mapping \( G \) to \( EFFECTED \). If this condition is satisfied, the relation in (16), parallel to (13), obtains:

(16)  \[ [ V [RP na- ] ] \|_{g_1} = \lambda x. \lambda e. \exists d[\text{process}(e) \land \text{INCREASE}(EFFECTED(x))(d)(e) \land d \in S_{EFFECTED} \land \text{max}(d)(S_{EFFECTED})] \]

Crucial is the next step. Matching (16) with the lexical component in (10a) creates the empty set, (17), since no object can be eaten and maximally effected in the same event.

(17)  \[ (9a) \cap (15) = \emptyset \]

What happens next is known from the literature on coercion. If combining two meanings leads to trouble, one of them gets re-analyzed in order to obtain a coherent interpretation. Following recent ‘constructionalist’ literature (see especially Borer 2005), I suggest that structural aspects of meaning are more sustainable while idiosyncratic ones are more flexible, as captured in (18):

(18)  Structural meaning preservation (Struct >> Lex)
Whenever Struct and Lex cannot match to yield an expression with a non-empty extension, Struct is preserved as much as possible.

According to (18), whenever the derivation is in danger of creating the empty set, as in (17), a lexical relation between individuals and events is coerced. I propose that this is done by existentially binding the individual variable, (19a). (19b) is thus a predicate of events in which something has been eaten.6

(19)  a.  Lex \rightarrow \exists \lambda \exists[Lex]
 b.  \exists \lambda \exists(\lambda x. \lambda e. \text{eat}(x)(e)) = \lambda e. \exists x[\text{eat}(x)(e)]

This predicate, then, can be integrated into the event structure as a condition on the event variable. We take the set of pairs of events and individuals from the extension of (16) and add the requirement that the events are identified as eating events from (19b)). This rule of construal is known as Event Identification (EI) from Kratzer 1996:

---

6One can think of other ways of how the relation \( \lambda x. \lambda e. \text{eat}(x)(e) \) can be readjusted. For one, the extension of \text{eat} can be expanded to include not just pairs of events and eaten objects but other types of pairs as well. After this expansion happens, the extension of \text{eat} will be large enough to intersect with (15) in a non-trivial way. The reason why I doubt that this is the right solution is: it is not obvious how the mechanism of expansion can be constrained. For the expanded \text{eat}, we have to admit pairs of events and effected individuals in its extension. If nothing else is said, we may end up having, as a limit case, every lexical verb denoting any possible relation between individuals and events.
Sergei Tatevosov

(20) a. \( \text{Match}(\text{Struct, Lex}) = \text{Struct} \cap_{EI} \exists_i[\text{Lex}] \)
    b. \( R \cap_{EI} P = \{ <x, e> | <x, e> \in R \land e \in P \} \)

(20a) is the rule deriving a coerced interpretation of ES. By hypothesis, it is invoked if and only if the regular rule in (5) fails to assign a non-empty extension to an expression under a particular choice of LIs. After (16) and (10a) combine via (20a), the outcome is the relation between maximally effected individuals and events in which something has been eaten in (21).

(21) \( \text{Match}(|| \text{ed} || g_1, || \text{V} \text{[RP na-]} || g_1) = \lambda x.\lambda e. \exists d \ [ \text{process}(e) \land \text{INCREASE(EFFECTED}(x))(d)(e) \land d \in S_{\text{EFFECTED}} \land \text{max}(d)(S_{\text{EFFECTED}}) \land \exists y[\text{eat}(y)(e)]] \)

This is the meaning of (2c), where \text{eat} is integrated into the ES characteristic of verbs of creation.

6. Activity event structure

The last substantial ingredient of the proposal aims at accounting for why we do not find arguments like \textit{the teapot} in or \textit{a belly} in (1c) and (2a).

Under the current set of assumptions, absence of arguments like \textit{the teapot} or \textit{a belly} amounts to unavailability of any version of the derivational scenario outlined in the previous section. In (2a), coercion of \( \lambda x.\lambda e. \text{eat}(x)(e) \) cannot happen.

To derive this, I only need two additional assumptions. First, the activity event template differs from the accomplishment event template in that it does not contain RP, (22) (see Ramchand 2008 for motivation).

(22) \([\text{VP} \text{DP} \quad [\text{V} \text{ V }]]\)   \hspace{1cm} \text{ACTIVITY EVENT TEMPLATE}

Secondly, the interpretations of V within accomplishment and activity ESs are not identical. As we saw, the semantic representation of accomplishment ESs contains the variable over gradable properties \( G \). Because of \( G \), accomplishments describe changes, represented above in terms of the \text{INCREASE} relation, and are endowed with a degree argument, which makes a prefix originating as the complement of \text{V} obligatory.

At this point, it is natural to follow the logic of Dowty (1979) and much subsequent literature that suggests that accomplishments are nothing but a sum of an activity and a change of state (“become”) components. Subtracting a change of state component from the denotation of \( V_{\text{ACC}} \) yields (23) as the structural meaning of activities:

(23) \( || V_{\text{ACT}} || = \lambda x.\lambda e. \text{process}(e) \land \text{arg}(x)(e) \)

In this impoverished, underspecified templatic meaning, an individual argument is no longer an entity that undergoes change along a specific dimension. All that is specified about this individual is that it stands in the maximally general \text{arg} thematic relation to an event (‘x is a participant of e’). Activities, unlike accomplishments, do not impose non-
Non-subcategorized arguments

trivial restrictions on the content of a relation between individuals and events. The crucial consequence of this underspecified character of activities is captured in (24):

(24) For any lexical verb $V_{\text{Lex}}$, $|| V_{\text{Lex}} || \subseteq || V_{\text{ACTIVITY}} ||$

From (24), it follows that the intersection of the relation introduced by a lexical verb and the structural relation denoted by $V_{\text{ACT}}$ is never an empty set (assuming that lexical verbs never have an empty extension). This means that no lexical verb will be ever coerced into an event predicate, (19), and integrated into an activity ES by Event Identification, along the lines of (20). The analysis predicts, correctly, that that an argument of an activity ES will also be an argument of $V_{\text{Lex}}$, for any $V_{\text{Lex}}$.

For (2a), in particular, combining the denotation of eat from (10a) with (23) via Match yields (25):

(25) $\text{Match}(|| ed ||, || V ||) = \lambda x. \lambda e. \text{process}(e) \land \text{arg}(x)(e) \land \text{eat}(x)(e)$

(25) is a relation between eating events and individuals eaten in those events, as required. This completes the outline of the two-level theory of argument projection developed in this paper. I am in the position of summarizing main results of the study.

7. Summary and conclusions

The theory I have outlined above seems to have achieved two related goals. First, it determines the conditions under which arguments not subcategorized for by a verb stem are projected. Secondly, it accounts for the observation from Section 1: sentences with non-subcategorized arguments entail that those arguments are not just holders of a result state, but also undergoers in a corresponding change of state subevent. Relying on the two-level architecture of ES, I have argued that in accomplishment ESs, the structural argument is always projected. If inserting a lexical verb into ES leads to a coherent interpretation, whereby a non-empty relation between individuals and events obtains, the resulting argument is understood as a ‘subcategorized’ one. If insertion creates an empty set, the lexical relation denoted by a verb gets reinterpreted through existential closure and turns into a predicate of events. The projected argument appears to be non-subcategorized. In that way, the condition on non-subcategorized arguments has been reduced to the semantic mismatch between the semantic content of lexical and structural components of ES.

The fact from Section 1 that a non-subcategorized argument ends up being a participant of a process that culminates in bringing about a result state is no longer puzzling. In the system developed here, it is an argument of such a process by virtue of being an argument of a gradable property that represents change in the course of an event. As soon as this move has been made, the puzzle dissolves.

To conclude, I hope to have developed a proposal that captures regularities underlying argument projection patterns in a less stipulative manner than other existing theories. Wider theoretical implications of the proposal remain to be tested against a wider array of argument projection phenomena.
References


Sergei Tatevosov
tatevosov@gmail.com