Romanian pluractional adverb *tot* and evaluation plurality[1]

Mara PANAITESCU - *University of Bucharest & CNRS-LLF, Université Paris 7*

Abstract. The paper proposes an explanation of the event internal – event external distinction within the class of pluractionals by resorting to the notions of encapsulation and decomposition, as defined in Dynamic Plural Logic. Two Spanish pluractional periphrases, incremental *ir* and frequentative *andar*, will be placed in the first category, while Romanian pluractional adverb *tot* will be placed in the second.

1. Introduction

The focus of this paper is a comparative analysis between two Spanish pluractional periphrases presented in Laca (2006)[2], frequentative *andar* (2) and incremental *ir* (4), and the Romanian pluractional adverb *tot* (1), (3).

(1) Tot deranjează lumea.
TOT disturb.PR peopleDEF.
'S/he keeps disturbing people.'

(2) Anda molestando a la gente.
Walk.PR.3S disturbing to the people
'S/he is giving people trouble.'

(3) Vremea se tot încălzea.
weatherDEF REFL TOT warmedIMPF
'The weather kept getting warmer.'

(4) La situación iba empeorando.
the situation go.IMPF worsening
'The situation was getting worse.'

The comparison is meant to provide a more general basis of classification of pluractional markers, one which expresses the event internal-event external distinction as a distinction between evaluation singularity and evaluation plurality.

To begin with, the common properties of the three pluractionals reflect their syntactic status as V^0 modifiers[3]. Firstly, all three distribute bare plural internal arguments into the iterated events. For instance, the use of a one-time-only achievement in (5a) has only one possible interpretation, one in which the fox killed a hen then a different hen, and so on. A meaning along the lines that the fox killed a plurality of hens and then repeated the action on the same group of hens is obviously excluded. The same is true for (5b) and (6).

(5) a. El zorro anduvo matando gallinas.
the fox walk.SP killing hens.
'The fox has been killing hens.'

---

[1] Many thanks to Brenda Laca, Carmen Dobrovie Sorin, Giorgio Magri, Marta Donazzan and all other participants in the Labex PLU seminar for helpful suggestions.
This work was possible with the financial support of the Sectoral Operational Programme for Human Resources Development 2007-2013, co-financed by the European Social Fund, project number POSDRU/107/1.5/S/80765

[2] All Spanish examples and judgments are from this source.

[3] This pattern is not an isolated case for pluractionals (see van Geenhoven (2004) for an analysis of the West Greenlandic frequentative pluractional *qattar* and of English temporal *for*-contructions).

b. Con el tiempo, el club fue perdiendo socios.
With the time, the club go.SP losing members
'The club gradually lost members.'

(6) Tot explodau bombe.
TOT explode.IMPF bombs
'Bombs kept exploding.'

The second important unifying feature of the three pluractionals is the impossibility of multiplication of the singular indefinite. (7a) is uninterpretable because it conveys that the same chicken was killed repeatedly. The same applies to (7b) and (8).

(7) a. ?? El zorro anduvo matando una gallina.
the fox walk.SP killing a hen
'The fox has been killing a hen.'
b. ?? Con el tiempo, el club fue perdiendo un socio.
with the time, the club go.SP losing a member
'The club gradually lost a member.'

(8) a. ?? *Ion a tot spart un pahar.*
Ion has TOT broken a glass.
?? 'Ion kept breaking a glass'.

Turning to the contrasts in the pattern of distribution, the Spanish pluractionals *andar* and *ir* don't license sentence-internal readings of *the same* (9), meaning that the book in (9), for instance, has to be contextually retrieved (the same book that Peter was reading, for example). Romanian *tot*, on the other hand, does license sentence-internal readings of *the same*. (10) may mean that Maria read (from) a book which is discourse-new, then (from) the same book, and so on.

(9) a. María anda leyendo el mismo libro.
Maria walk.PR reading the same book
'Maria has been reading the same book [on and off].' (**sentence-external only**)

b. Juan fue calentando la misma sopa.
Juan go.SP heating up the same soup
'Juan gradually heated up the same soup.' (**sentence-external only**)

(10) Maria tot citeşte aceeaşi carte.
Maria TOT reads the-same book
Mary keeps reading the same book. (**sentence-internal OK**)

The second contrasting property is that *tot* doesn't display distributional effects with universally quantified objects (12), *andar* and *ir* do (11). (12) cannot mean that Ion called his friends one by one until he called them all, it may only mean something strange as calling all friends, then doing it again many times.

(11) a. Juan ha andado llamando por teléfono a cada uno de sus amigos.
Juan has walked calling by phone to each one of his friends
'Juan has been phoning every one of his friends.' **= each friend may be called once**
b. Con el tiempo, el club fue perdiendo (a) todos sus socios.
With the time, the club go.SP losing (to) all its members.
'As time went by, the club gradually lost all its members.'

(12) ??Ion a tot sunat toți prietenii.
Ion has TOT called all friends.DEF
?? 'Ion kept calling all of his friends.' **= each friend was repeatedly called**;

Having looked at two unifying and two contrasting distributional facts[4], an explanation for these patterns of interaction with internal arguments cannot be due to scope dependencies in either of the three cases. The Spanish pluractionals are syntactically lower than viewpoint-aspect modifying *estar*[5]. *Tot* is also necessarily lower than all other verbal material (Tense, Aspect, other adverbial and pronominal clitics);

(13) Se poate să (*tot) fi **tot** încercat să ne contacteze cât am fost plecați.
REFL can CONJ *TOT PRF **TOT** tryPART CONJ us$_{CL}$ contact as-much have1PL been gone
'S/he may have kept trying to contact us while we were gone.'

Instead, I will resort to an account within Dynamic Plural Logic (DPlL) in order to find an explanation for the common features, as well as the contrasts. The reason for choosing dynamic logic is that it distinguishes between two types of plurality: domain plurality and evaluation plurality. In the absence of scopal dependencies between the pluractional and the internal argument, the data above suggest that the two types of plurality are responsible for the contrasts observed.

The distinction between domain and evaluation plurality is employed, among others, in Brasoveanu & Henderson (2009). In this account, *one by one* expresses an evaluation singularity, therefore no sentence-internal readings of *different* are observable. The mechanism responsible for the distributivity effect is called **ENCAPSULATION**. *Each*, on the other hand, is an evaluation plural distributor and sentence-internal readings of *different* are licensed. The mechanism is called **DECOMPOSITION**.

In the present analysis, I assume that *andar* and *ir* store a plurality of events in a single value assignment (they are evaluation-singular) and establish **thematic dependencies** by **encapsulation**. Conversely, *tot* introduces events distributively by **decomposition** (evaluation

[4] Henderson (2011) follows this line of reasoning for Kaqchikel pluractional affix *la'*.
[5] Laca (2006)

plurality) and imposes the constraint of **type identity** between them. Unlike quantifiers like *each* and *every*, the dependencies induced by these pluractionals are **scopeless**.

A consequence of this treatment is that it offers a new perspective on the event-internal (EI) – event-external (EE) distinction. Many authors link EI pluractionality with a "grouping together" of events, but based on different criteria[6]. In Cusic (1981), the relevant criterion is the level of complexity. EI pluractionality occurs at phase-level, while EE pluractionality is event/ occasion-level. In Lasersohn (1995), the criterion is the P-V (non-)identity, where P is the predicate that applies to the subevents of the V-event. EI pluractionals establish non-identity, as opposed to EE. Wood (2007) assumes EI, but not EE pluractionals, are formed via a group formation operator applied to events. Rothstein (2008) introduces the notion of S-cumulativity and the S-sum formation operator, which is responsible for turning semelfactives into activities. Under the view adopted here, the basic differentiation is at the level of evaluation: encapsulation is EI; decomposition is EE.

The paper is organized as follows: in section 2., I present the theoretical background, the notions of encapsulation and decomposition, and the application to quantificational dependencies. Section 3. is dedicated to the account of Spanish *ir* as involving encapsulation via an incremental function. Section 4. deals with the encapsulation procedure for Spanish *andar*. In section 5. I analyze Romanian *tot* as involving decomposition and verification of type identity between events.

2. The framework (DPlL)

The Spanish and Romanian pluractionals are analyzed using the theoretical background in Henderson's (2011)[7] account of the distributive pluractional *la'* in Kaqchikel, couched in a version of DRT called Dynamic Plural Logic (DPlL), as developed by van den Berg (1996), Nouwen (2003), Brasoveanu (2010). Besides the domain of truth values, the model contains a domain of individuals D_e, which is the powerset of a designated set of entities IN minus the empty set: $\wp^+$(IN). $\wp^+$(EV) is the domain of events D_s (the powerset of a designated set of events EV minus the empty set). The "part of" relation $\leq$ over individuals/ events is set inclusion over $\wp^+$(IN) /$\wp^+$(EV) such that $a \leq b$ iff $a \subseteq b$. The sum operation $\oplus$ is set union over $\wp^+$(IN) /$\wp^+$(EV) such that $a \oplus b := a \cup b$.

2.1. The Semantics of the Logical Language

Interpretation is relative to **sets of assignment functions** $G = \{ g1, g2, g3, \cdots \}$, called contexts/ information states. $\langle G, H \rangle$ are input-output pairs of sets of assignments. If *i* is an index on a variable, and *G* is a set of assignment functions, then $G[i] = \{g(i) : g \in G\}$, the set of values that the assignment functions in G assign to variable i. Formulae describe relations between assignment functions. Relations (noted R) have no effect on the input:

[6] See Souckova (2009), Cabredo-Hofherr (2010) for an overview.
[7] Some of the theoretical issues and examples are extensions and discussions of Henderson (2011) in Cable (2012).

(14) $[[\ R(v_i \ldots v_n)\ ]]^{\langle G,H \rangle} = T$ *iff* $G = H$ and $[[R]]^{\langle g,h \rangle}(h(i))\cdots(h(n))$ for all $h \in H$;

The introduction of a new variable, on the other hand, requires an update of the input function:

(15) a. g[i]h
→ *h* is just like g except for what it maps index i to;

b. $[[\ [\ xi\ ]\ ]]^{<g,h>} = T$ iff g[i]h;

c. *G*[i]*H*
→ for every $g \in G$, there is an $h \in H$ such that *g*[i]*h*, and for every $h \in H$, there is a $g \in G$ such that *g*[i]*h*;

d. $[[\ [\ xi\ ]\ ]]^{<G,H>} = T$ *iff* G[i]H;
→ *G*[i]*H* holds *iff* all the assignment functions in *G* and *H* look exactly the same, except in their value for the index *i*;

2.2. Two types of plurality: domain-level and evaluation-level

The main advantage of DPlL as opposed to traditional dynamic frameworks working on single assignments is the possibility of distinguishing between two types of plurality. In DPlL terms, domain-level plurality depends on individual cells of the matrix in a context[8]. Domain plurality is determined by checking whether an assignment h in H maps a variable to a singular individual or plural individual. Evaluation plurality, on the other hand, depends on a column of the matrix and is determined by checking whether or not the assignments in H map a variable to more than one individual across a column. Domain singularity is verified by the predicate **atom**, while domain-level plurality is signaled by specifying the cardinality of a given assignment function, e.g. x_3 in (16) verifies the following condition[9]:

(16) $[[\text{two}(x_3)]]^{\langle G,H \rangle} = \mathbb{T}$ iff $G = H$ and for all $h \in H$, $|\{x' : x' \in h(3) \wedge \text{atom}(x')\}| = 2$;

Evaluation-singularity is expressed by the predicate **singleton**. Evaluation plurality is expressed by specifying the cardinality at the assignment-set level:

(17) $H(i) := \{h(i) : h \in H\}$

(18) $|H(i)|$ is the cardinality of the set of individuals $H(i)$;

An illustration is provided below, where *a* to *l* are individuals:

[8] See (19) below for a visual representation.
[9] Variables are identified by means of indices. See below in (19) that the variable indexed 3 does verify this condition.

(19)

H	x_1	x_2	x_3	x_4
$h1$	a	b	$e\oplus f$	$g\oplus h$
$h1$	a	c	$e\oplus f$	$i\oplus j$
$h3$	a	d	$e\oplus f$	$k\oplus l$

In (19), x_1 is domain-singular (atom) and evaluation-singular (singleton). x_2 is domain-singular (atom) and evaluation-plural (non-singleton - H(2) = {b, c, d}). x_3 is domain-plural (non-atomic) and evaluation-singular (singleton- H(3) = { $e\oplus f$ }). x_4 is domain-plural and evaluation-plural. The distinction between the two types of plurality proves to be relevant in teasing apart two types of quantificational dependencies: encapsulation (*one by one*) and decomposition (*every*), discussed in Brasoveanu & Henderson (2009).

The mechanism of **encapsulation into a function** involves **Skolem functions** which **store quantificational dependencies as a whole**, mapping each entity to the (possibly non-atomic) entity that depends on it. *One by one* breaks the event down into **temporally sequenced** subevents and it distributes the plural participant over these subevents. In (20), *one by one* associates each boy-atom with an event of reciting a poem and induces a dependency between boys and the recited poems encapsulated in a function *f* in (20b). Thus, encapsulation is storage in a single assignment *g* and involves domain plurality, but evaluation singularity.

(20) a. The boys recited a poem one by one.
b.

	x(boys)	*f*(boy-poem dependency)	
g	the.boys (=*g*(*x*))	boy1 → poem1 boy2 → poem2 etc.	=*g*(*f*)

Each, on the other hand, establishes dependencies via **decomposition** of the distributive quantification into sets of assignments, such that each *n*-tuple of quantificationally dependent entities is individually stored in a variable assignment. The distributor *each* breaks the plural individual *the.boys* into atoms and stores every boy-atom in a variable assignment: *boy1* in *g1*, *boy2* in *g2* etc. The rest of the sentence is **interpreted relative to each variable assignment**: in each assignment, i.e., relative to each boy, we store a possibly different poem.

(21) a. The boys each recited a poem.
b.

G	...	x (boys)	y (poems)	...	
g_1	...	boy_1 $(= g_1(x))$	$poem_1$ $(= g_1(y))$	...	boy_1 recited $poem_1$
g_2	...	boy_2 $(= g_2(x))$	$poem_2$ $(= g_2(y))$	...	boy_2 recited $poem_2$
...	...	...	...		...

A question which naturally comes to mind is why two distinct mechanisms are needed and where the two options make a noticeable difference in the interpretation. After all, encapsulation and decomposition don't behave differently with respect to cross-sentential anaphora:

(22) a. One by one, the boys chose a book.
b. Then, one by one, they opened it and read out the title.

(23) a. The boys each chose a book.
b. Then, they each opened it and read out the title.

But, as pointed out in Brasoveanu & Henderson (2009), they do differ with respect to licensing sentence-internal *different*:

(24) The boys each recited a different poem. (sentence external/ internal)

(25) The boys recited a different poem one by one. (sentence-external only)

Moreover, this is not an isolated phenomenon, particular to *one by one* and *each*. Generally**, sentence-internal readings require evaluation plurality and distributivity**. For Henderson 2011, the pairing of the two ingredients (evaluation plurality and distributivity may come about in two ways: by means of scope relations (in the case of distributive quantifiers like *each* and *every*) or in cases of scopeless dependencies (in the case of pluractionals). The next section is devoted to extending this generalization to the Spanish and Romanian pluractionals.

3. Incremental *ir* and encapsulation

To begin at an intuitive level, sentence (26) involves an incremental function mapping members of the club onto chronologically ordered losing subevents, as in (27). The subevents and their participants (the members) are stored in a single assignment function, therefore it is a case of evaluation singularity. The nature of the incremental function is discussed in the next subsection, after which a semantics for *ir* is proposed in 3.2.

(26) Con el tiempo, el club fue perdiendo (a) todos sus socios.
with the time, the club go.SP losing (to) all its members
'As time went by, the club gradually lost all its members.'

(27)

e	=	e1	⊕	e2	⊕	…
		LOSE(e1)		LOSE(e2)		
		runtime(e1)	<	runtime(e2)		
		incr(e1)		**incr(e2)**		
all members	=	member1	⊕	member2	⊕	…

3.1.The incremental function

Informally, (27) means that the club lost more and more members until it lost them all. The incremental function imposes constraint on the subevents in the following way: the predecessor function *pred[10] picks out the unique sum event *e'* which precedes an event *e* and satisfies the predicate[11]:

(28) $*pred(e) = \iota e'.\ \tau(e') < \tau(e) \wedge \forall e'' [\tau(e'') < \tau(e) \rightarrow e'' \leq e']$

The incremental function[12] compares two events with respect to a gradable property of events (type ⟨s,⟨⟨s,dt⟩,⟨s,t⟩⟩⟩) and says that an event *e* posseses this property to a greater degree than all its predecessors;

(29) $incr(e) = \lambda e.\lambda R\langle s,\langle dt\rangle\rangle.max(R(*pred(e)) > max(R(*pred(e) \oplus e);$

The function ***pred** sums up all predecessors of an event instead of simply comparing an event to its immediate predecessor. This is needed to include as possible values for R both monotonic and non-monotonic dimensions[13]. *Ir* may be mapped onto a **non-monotonic** dimension (one which doesn't track the part-whole relations that are relevant for the internal argument). Such a case would be (9b) in section 1, built with *ir* + *heating the soup*:

R = temperature;
max(R(e1)) = 20 degrees;
max(R(e1+e2)) = 25 degrees;
max(R(e1+e2+e3)) = 30 degrees;

In these cases, it would have been sufficient to compare the maximal degrees of e3 to that of its immediate predecessor e1. But *ir* may also be mapped onto a monotonic dimension, as in *ir* + *put on shirt, tie and jacket*:

R = quantity of things that are put on;
max(R(e1)) = shirt;
max(R(e1+e2)) = shirt+tie;
max(R(e3)) = shirt+tie+jacket;

Thus, *pred prevents a situation in which the shirt was put on, then removed, after which the tie was put on. **Incr** explains the contrast with *andar* and *tot* in (30) and (31):

(30) ??El río andaba creciendo.
The river walk.IMPF growing

[10] Inspired by Beck's (2012) implementation of the predecessor and sequence functions in the analysis of pluractional comparatives; the difference is that the predecessor is made up of the sum of preceding events; this is very close to her analysis of incremental *more* (see (175)-(177), p. 100-101);
[11] I leave aside the problem of the first predecessor to avoid complicating things too much; see Beck and von Stechow (2007) for a discussion.
[12] Bhatt and Takahashi (2008), in Beck (2012).
[13] See Schwarzschild (2006) for a discussion of the contrast between expressions like *20 degree water* vs. *twenty liters **of** water*.

??'The river was rising on and off.'

Laca (2006) comments that the only interpretation for (30) "is one attributing a sort of erratic behavior to the river, its level rising, going down again, rising again and so forth".

(31) Ion şi-a tot pus cămaşa şi geaca.
Ion REFL-has TOT put shirt.DEF and jacket.DEF
= Ion kept putting on his shirt and jacket.
≠ Ion put his shirt and jacket successively.

(31) says that Ion repeated the following action many times: putting on his shirt and jacket, which only makes sense if he also took them off each time. One may conclude that *tot* does not introduce an incremental function by itself. The incremental readings are present with degree achievements only as in (3), translated as 'The river kept rising'.

3.2. *Ir* and encapsulation

Coming back to our initial example in this section, it has already become apparent that, in the presence of a direct object, the incremental function traces the event-incremental theme dependency:

(32) Con el tiempo, el club fue perdiendo (a) todos sus socios.
with the time, the club go.SP losing (to) all its members.
'As time went by, the club gradually lost all its members.'

I will apply the same reasoning as in Brasoveanu & Henderson (2009) for *one by one*:

(33) a. One by one, the boys recited a poem.
b. One by one, the boys chose a different poem
→ sentence-external only

The fact that *one by one*, unlike *each*, doesn't license sentence-internal readings leads to the conclusion that the boy-poem dependency is stored as a whole. The same applies to *ir*, as attested by the unavailability of sentence-internal readings with *the same* in (34). The losing events - members dependency is encapsulated (introduced by a single assignment function g, as in (35).

(34) Juan fue calentando la misma sopa.
Juan go.SP heating up the same soup
'Juan gradually heated up the same soup.'
→ sentence-external only

(35)

	e	f(lose-member dependency)
g	*LOSE(=g(e))	{ e1 → member1 e2 → member2 … }

The function **incr** targets the number of participants such that linearly ordered atomic subparts of the event are mapped onto atomic subparts of the plural participant introduced by the θ-function. The information state stores quantificational dependencies (between events and participants) as a whole in a function.

(36) $ir \rightarrow \lambda E_{st}.\lambda_{es}.\ E(e) \wedge \text{linear.order}(\{e' \leq e\}) \wedge |\{e' : e' \leq e\}| > n \wedge$
$\forall e' \leq e\ (\text{incr}(e'))$;

I mention some of the advantages of assuming (36) as a description of the contribution of *ir* to the interpretation of predicates of events. Firstly, the dependency between events and their corresponding incremental themes is established indirectly, by means of the **incr** function. Secondly, the dependency is stored in a single assignment (domain-plurality but evaluation singularity), which prevents the multiplication of the singular indefinite (see (7b)). Thirdly, a desirable outcome is that **incr** is not prevented from mapping onto parts of the object:

(37) a. María fue leyendo La Guerra y la Paz .
María go.SP reading The War and the Peace.
'María gradually read *War and Peace.*'
b. R = page number;
e1= page 20
e2 = page 40 etc.

4. Frequentative *andar* and encapsulation

Intuitively, (38) is interpreted along the lines of (39): Juan called a friend then another, until all friends were called.

(38) Juan anda llamando por teléfono a cada uno de sus amigos.
Juan walk.PR calling by phone to each one of his friends
'Juan is phoning every one of his friends.

(39) e = e1 ⊕ e2 ⊕ …
CALL(e1) CALL(e2)
runtime(e1) < gap < runtime(e2)
each friend = friend1 ⊕ friend2 ⊕ …

Moreover, *andar* does not license internal readings for *the same* (see (9a)), which suggests that *andar* is not distributive over events in the same way that the universal quantifier *each* is over individuals.

(40) María anda leyendo La Guerra y la Paz.
*María walk.*PR *reading The War and the Peace*
'María is/ has been reading *War and Peace* [on and off].'

Looking at (40), the event-theme dependency seems very loose. First of all, there is no measure function that maps reading subevents onto parts of the book, which corresponds to the fact that the subevents (of reading parts of *War and Peace*) are interpreted as atelic, and so is the resulting superevent. Remember that there are no multiplication effects with singular indefinites either (see (7a)), such that substituting *War and Peace* with *a book* in (40) produces the same atelic interpretation: Maria has been reading from a book. Bare plurals, on the other hand, are distributed in the subevents (see (5a)), such that substituting *War and Peace* with *books* in (40) produces a reading in which Maria reads a book, then a different one and so on. The conclusion to be drawn is that the dependency event-theme is underspecified (unlike for *ir*). An indication that this intuition is on the right track is the behavior of *andar* with degree achievements (the river going up and down in (30)). Formally, the interpretation of the pluractional looks as in (41):

(41) *andar* $\rightarrow \lambda E_{st} .\lambda_{es} .\ E(e) \wedge \text{linear.order}(\{e' \leq e\}) \wedge |\{e': e' \leq e\}| > n$
$\wedge\ \forall e', e'' \leq e\ (\exists t\ (\tau(e') < t < \tau(e'') \vee \tau(e'') < t < \tau(e'))) \wedge$
$\boldsymbol{\forall e' \leq e\ (\exists x \leq th(e)\ [x = th(e')]))}$;

The conditions introduced by *andar* are the following (line 1, line 2 and line 3 in (41)): the first line requires temporal sequence; the second, temporal gaps; the third, an **underspecified thematic dependency**.

The underspecified temporal dependency predicts that one-time achievements require that x = th(e), hence the oddity in the interpretation of (42). In DRT, a singular indefinite simply introduces a new variable x and the conditions **singleton(x)** & **atom(x)**, which means for the variable **error(x)** in (42) that the same error is found over and over:

(42) ??María andaba descubriendo un error en el manuscrito.
*María walk.*IMPF *discovering a typo in the manuscript*
??'Maria was discovering a typo in the manuscript.

(43)

	e	f(find-error dependency)
g	*FIND(=g(e))	{ e1 → error1 e2 → error1 … }

Nevertheless, the behavior of the underspecified thematic relation with a universally quantified object is still in need of an explanation.

(44) Juan anda llamando por teléfono a cada uno de sus amigos.
Juan walk.PR calling by phone to each one of his friends
'Juan is phoning every one of his friends.'

I adopt the translation for *each* in Brasoveanu&Henderson (2009).

(45) $\text{each}^{th} \rightarrow \lambda X_{et}.\lambda E_{st}.\lambda e_{s}.\forall e' \leq e\ (\mathbf{atom(e') \rightarrow E(e') \wedge th[\{e' \leq e: atom(e')\}] = \{x \in X: atom(x)\}}$
where $th[E] = X =^{def} X = \{th(e): e \in E\}$;

(46) *andar calling each friend*:
(i) $\exists e\ (\forall e' \leq e\ (atom(e') \rightarrow CALL(e') \wedge$
(ii) $th[\{e' \leq e: atom(e')\}] = \{x \in FRIEND: atom(x)\} \wedge$
(iii) $linear.order(\{e' \leq e\}) \wedge |\{e': e' \leq e\}| > n \wedge \forall e', e'' \leq e\ (\exists t\ (\tau(e') < t < \tau(e'') \vee \tau(e'') < t < \tau(e'))) \wedge$
(iv) $\forall e' \leq e\ (\exists x \leq th(e)\ (x = th(e')))$;

The first line says that there is an event *e* composed of atomic subevents of calling. The second line says that the set composed of the themes of each subevent is the set of all atoms in the predicate FRIEND. The third line says that the subevents are linearly ordered, have a cardinality greater than n and are separated in time. Finally, the fourth line says that each subevent is related to a part of the theme of its superevent.

5. *Tot* and decomposition

5.1. Theoretical background: operators max, dist and concatenation

I will first present Brasoveanu's (2011) alysis of the mechanism for licensing internal readings of *the same* and *different* by nominal quantifiers. Just as in Dynamic Plural Logic (van den Berg (1996); see also Nouwen (2003)), **information states** I; J; etc. are modeled as sets of stacks (rows) {i1; i2; i3; . . .}; {j1; j2; j3; . . .}; etc. Plural info states enable us to encode discourse reference to: **quantifier domains** stored in columns (such that each column has an index); **quantificational dependencies** stored in rows (⟨a1,b1,c1⟩, ⟨a2,b2,c2⟩…).

A difference between what we have seen so far is that, here, expressions are interpreted relative to pairs of info states (instead of single info states ⟨I; J⟩). For example, ⟨I; K⟩, ⟨J; K'⟩ is an input-output sequence of pairs of info states. The right-side info states (K, K') are only used for sentence-internal dependencies. Their use is to temporarily form new info states by **concatenating** stacks from the initial (left-side) info state. The effect of concatenation is that elements of the quantificational domain (stacked vertically) are temporarily cast in a new stack (horizontally), which allows for quantificational dependencies to be established.

(47) identical $\{u_n, u_{n'}\} := \lambda\langle I,K\rangle . I_{un \neq \#, un' \neq \#} \neq \emptyset \wedge$
$\{x \leq \oplus u_n I\text{: atom}(x)\} = \{x' \leq \oplus u_n I\text{: atom}(x')\}$;

(48) $\text{same}^{m}_{un} \rightarrow \lambda P_{et}. \lambda v_e. P(v); {*}(P(u_{n+m}); \text{identical}(u_{n+m}, u_n)$;

(49) $[[\text{ Max i},\varphi\ ]]^{<G,H>} = \mathbb{T}$ *iff* $[[\ [x_i]\ \&\ \varphi\ ]]^{<G,H>} = \mathbb{T}$ AND
for all H', if $[[\ [x_i]\ \&\ \varphi\ ]]^{<G,H>} = \mathbb{T}$, then $H'[i] \subseteq H[i]$

- requires that *H* differ from *G* in the values it assigns to index i;
- requires that $[[\varphi]]^{<G,H>}$ hold (the conditions introduced in the DRS);
- H[i] is the *largest* set of entities that satisfy the conditions set by φ

(50) $\text{dist-WHOLE}_{un}\ (D) := \lambda\langle I,K\rangle . \lambda\langle I,K\rangle ; K = K' \wedge$
$u_n I = u_n J \wedge I_{un = \#} = J_{un = \#}$
$(|u_n I| = 1 \rightarrow D\langle I_{un \neq \#}, K\rangle, \langle J_{un \neq \#}, K\rangle \wedge$
$(|u_n I| \geq 2 \rightarrow \forall x \in u_n I(D\langle I_{un = x}, J_{un \neq \#}\rangle\langle J_{un = x}, J_{un \neq \#}\rangle$

The last line in (50) is the most important and says that if there are at least two rows storing different individuals in I for the variable u_n, then the left input info state stores in turn every x in u_n, while the right input and output states stores the entire column u_n, including the stacks that assign x as a value. This will serve as the basis for comparison for the adjective *same*, which introduces the **concatenation operator ***. (51) is an illustration of the procedure:

(51) a. *all* All^{u0} boys read the^{u1} same^{2}_{u1} poem.

b. *all* → $\lambda P_{et}. \lambda P'_{et}$: $\mathbf{max}^{un}$ ([atoms-only$\{u_n\}$; P(u_n)); $\mathbf{dist\text{-}WHOLE}_{un}$; (P'($u_n$)

c. $\underline{\mathbf{max}^{uo}}$ ([atoms-only$\{u_0\}$; boy$\{u_0\}$; max^{u1} ([atoms-only$\{u_1\}$]; poem$\{u_1\}$);
$\underline{\mathbf{dist\text{-}WHOLE}_{uo}}$(*([identical$\{u_{1+2}, u_1\}$]))); [singleton$\{u_1\}$]; [read$\{u_0, u_1\}$];

d.

All(u_o) boys — dist (recited the same poem)

u_o
b1
b2
b3

u_o	u_1
b1	p1

*

u_o	u_1
b1	p1
b2	p1
b3	p1

-->

u_o	$\mathbf{u_1}$	u_2	$\mathbf{u_3}$
b1	**p1**	b1	**p1**
b1	**p1**	b2	**p1**
b1	**p1**	b3	**p1**

& g(u_1)= g(u_3) for all g

u_o	u_1
b2	p1

*

u_o	u_1
b1	p1
b2	p1
b3	p1

-->

u_o	$\mathbf{u_1}$	u_2	$\mathbf{u_3}$
b2	**p1**	b1	**p1**
b2	**p1**	b2	**p1**
b2	**p1**	b3	**p1**

& g(u_1)= g(u_3) for all g

The conclusions so far are that whole-set based distributivity pairs each individual with all the individuals in the set we distribute over, including itself; this excludes the possibility of licensing sentence-internal *different*, which is unavailable for *tot*, but allows the licensing of sentence-internal *same*, as desired.

5.2. Pluractional adverb *tot* and decomposition

I will consider that the semantic effect of the pluractional adverb *tot* is roughly equivalent to the expression *again and again*. The adverb *again*[14] is anaphoric on a previously introduced event-type discourse referent[15]:

(52) a. John^{u0} sighed^{e1}. He_{u0} sighed^{e2} again_{e1}.

b. $[[\text{again}]](E_{\langle s,t\rangle})(e) = 1 \text{ iff } P(e) \ \&\ \exists e'[e' < e \ \&\ P(e')]$

Subscript 1 on *e* identifies the event with which e2 is to establish an anaphorical relation. A*gain and again*, on the other hand, licenses a sentence-internal dependency between events:

(53) John^{u0} sighed^{e1} again and again^{+1}_{e1}.

Superscript +1 indicates the distance between the first element of comparison (e1) and the second member (e1+1= e2, which is the position where the values e1 are stored via concatenation). Assuming that *tot* is a V^0-modifier, we have the following order of discourse referents (54a). *Tot* introduces evaluation-level plurality by storing subevents in separate stacks (**DECOMPOSITION**).

(54) a. Maria^{u0} ${}^{e1}[\text{TOT}^{+1}_{e1}$ sighs].

b.

G	u0	e1
g1	Maria	sigh1
g2	Maria	sigh2
g3	Maria	sigh3
…	…	…

The semantic contribution of *tot* is then looks as in (55)[16]:

(55) $tot^{+1}_{e1} \rightarrow \lambda E_{st}\,.\lambda e_{s.;}\ \max^{e1}(E\{e1\};\ |e1| > n;\ \text{dist-WHOLE}^{e1}$ (*([identical{type(e1+1),

[14] I am only interested in the repetitive reading, not in the restitutive one (see Beck(2005)), because only the former is involved in the *again and again* construction.

[15] Superscripts introduce new discourse referents; subscripts refer to previously introduced referents; I use *u* and *e* for individuals and events; the indices keep track of the order in which discourse referents are introduced.

[16] This is more a description of the mechanism. I do not take a stand as to what type identity for events means in this paper (see next subsection for discussion).

type(e1)}]));

Max^{e1} introduces the maximal set of sighing events that verify the conditions: e1 satisfies the predicate E. The cardinality of e1 is greater than a contextually set n. By dist-WHOLE, the events are stored by different assignment functions and verify the conditions in the DRS one at a time (evaluation plurality, see (54b)). Concatenation compares each event in turn with the values others stored in e1 (including itself) and verifies whether they are of the same type.

5.3 *Tot* and event type identity

This subsection offers some motivations for the type-identity condition. I begin with the observation that *tot* is distributive, but not necessarily down to atomic events. For instance the Romanian translation of TOT + *jump on one foot seven times* can either mean that the jumping on one foot was done on seven separate occasions, or that seven jumps were performed numerous times. *TOT + call three friends* may mean that friends are called either individually or collectively.

So, distributivity down to atoms of both events and their corresponding themes, which is the key ingredient in Henderson (2011)'s account of pluractional *la'*, is not an option. Here, the superevent is structured by iteration of the same event type, no matter how complex:

(56) a. Maria tot citea o carte.
Maria TOT read.IMPF a book.
'Maria kept reading a book.'
= kept reading parts of the same book
= kept reading the whole book

The mention of event types invites to a discussion of the status o events, for which I will resort to the overview in Moltmann (2002) of two views on this topic. The most widely accepted view is that events are **primitive objects**, on a par with individuals (Davidson (1967)). Thus, the identity of events does not depend on properties. A problem for the Davidsonian view is that there is no straightforward procedure of reference to event types. In (57), the frequency adverbial counts the insantiations of an event type at particular times:

(57) The closing of the door takes place every day.

If events are taken to be arguments of verbs, they would have to anticipate not only the meaning of the verb, but also of its arguments and modifiers, which leads to complications in case more specific descriptions of the same event are added by means of adverbials. For example, *the careful closing of the door at twelve o'clock every day* would be a completely different event from the one in (57).

A different perspective is offered in Kim (1976), according to which events are derived objects and they are individuated entirely on the basis of individuals, properties and times. In this type of accounts, events are considered to be **property exemplifications**. Verbs don't express properties of events, but properties of individuals, on which events may then depend. For instance, according to Bennet (1988), events are introduced into the semantic structure by means of nominalizations ([John's walk] = f([walk], John, t). But, the adoption of the second type of accounts leads to a different problem, the fact that it misses the concreteness of events. Events, unlike facts, are grounded in specific objects having certain kinds of properties at specific times.

Without taking a general stand, I will consider that the identity condition that is part of the meaning of *tot* does verify type identity indirectly, in the manner envisaged in the property exemplification accounts.

6. General conclusion

The two unifying properties for the three pluractionals, mentioned in the beginning, were distribution over bare plurals (BPs) and impossibility of multiplication of the singular indefinite. Within a dynamic semantic framework, the explanation may be derived from three assumptions. Firstly, both BPs and singular indefinites introduce a free variable. Secondly, BPs are number-neutral, as proposed in Zweig (2009). Thirdly, singular indefinites come with the conditions **singleton** and **atom** on the newly-introduced variable. The condition "atom" prevents domain-level plurality, while the condition "singleton" prevents evaluation plurality. Given that the pluractionals establish scopeless dependencies, they cannot "multiply" a singular indefinite[17] by means of the same procedure as quantifiers like *each*.

Regarding the interaction with inner aspect, a number of generalizations can be derived. Firstly, pluractionals which distribute over events and participants via encapsulation display a group-like behavior and may be ambivalent with respect to telicity. Secondly, decomposition requires atelicity at the global level, even though each subevent may specify complete, telic events. In this case, we are dealing with an instance of event plurality at the level of evaluation. From the present perspective, the group-like interpretation of EI pluractionals and the complexity of EE pluractionals follows naturally from the type of mechanism responsible for value assignment. Finally, the generalization should be tested on other pluractionals as well. For instance, English temporal *for*-constructions and *keep V-ing* are expected to pattern like Romanian *tot*:

(58) a. John drank beers in five minutes for an hour[18].
b. John kept drinking beers in five minutes (till he got sick).

[17] A dependent indefinite such as Romanian *câte*, on the other hand, imposes a non-singleton condition:

(i) Maria a tot citit câte o carte.
Mary has TOT read CÂTE a book
= Mary kept reading a book after another.

[18] These types of examples are discussed in MacDonald (2008).

References

Beck, S. (2005). There and back again. *Journal of Semantics* 22:3-51.

Beck, S., and von Stechow, A. (2007). Pluractional adverbials. *Journal of Semantics*, 24(3), 215-254.

Beck, S. (2012). Pluractional Comparisons. *Linguistics and Philosophy*, 35:57–110.

Bennet, J. (1988). Events and their names. Oxford: Clarendon Press.

van den Berg, M. (1996). Some aspects of the internal structure of discourse: ILLC, University of Amsterdam dissertation.

Bhatt, R. and S. Takahashi (2008). When to reduce and when not to: crosslinguistic variation in phrasal comparatives. In Glow 2008, Newcastle University, Newcastle.

Brasoveanu, A. and R. Henderson (2009). Varieties of distributivity: One by one vs. each http://people.ucsc.edu/~abrsvn/varieties_of_distributivity_SALT_19.pdf

Brasoveanu, A. (2010). Plural discourse reference. In van Benthem, J. and A. ter Meulen (eds.), *Handbook of logic and language*, 2nd edition, Elsevier.

Brasoveanu, A. (2011). Sentence-Internal Different as Quantifier-Internal Anaphora, http://people.ucsc.edu/~abrsvn/different_revised_v1.pdf

Cable, S. (2012). Distributive Numerals and Pluractionals in Kaqchikel and DPlL: Henderson (2011). For "Distributivity, Pluractionality and Reciprocity Across Languages" (Semantics Seminar; Spring 2012); http://people.umass.edu/scable/LING753-SP12/Handouts/17.Henderson-2011.pdf

Cabredo Hofherr, P. (2010). Verbal plurality and event plurality. Course material, Summer School on Linguistic Typology, Leipzig, 15/08 – 3/09/2010. http://www.eva.mpg.de/lingua/conference/2010_summerschool/pdf/course_materials/Cabredo2010p lV.pdf

Cusic, D. (1981). *Verbal plurality and aspect*. PhD dissertation, Stanford University.

Davidson, D. (1967). The logical form of action sentences. In N. Rescher (ed.) *The logic of decision and action*. Pittsburgh: Pittsburgh University Press.

van Geenhoven, V. (2004). For-adverbials, frequentative aspect and pluractionality. *Natural Language Semantics* 12, 135-190.

van Geenhoven, V. (2005). Atelicity, pluractionality, and adverbial quantification. In: Verkuyl, H., H. de Swart and A. van Hout (eds.), *Perspectives on aspect*. Dordrecht: Springer, 107-124.

Henderson, R. (2011). Pluractional distributivity and dependence, SALT 21.

Kim, J. (1976). Events as property exemplifications. In M. Brand & D. Walton (eds.) *Action Theory*. Dordrecht: Reidel.

Laca, B. (2006). Indefinites, quantifiers and pluractionals: What scope effects tell us about event pluralities. http://www.umr7023.cnrs.fr/sites/sfl/IMG/pdf/LacaIndPlurac-fin.pdf

MacDonald, J.E. (2008). *The Syntactic Nature of Inner Aspect: A Minimalist Perspective*. Linguistik Aktuell/Linguistics Today series. Amsterdam: John Benjamins.

Moltmann, F. (2002). 'Events as Derived Objects'. Proceedings of the *Colloque de Syntaxe et Semantique*, Paris 2001 (CSSP01).

Nouwen, R. (2003). Plural pronominal anaphora in context: UiL-OTS, Utrecht University diss.

Rothstein, S. (2008). Two puzzles for a theory of Lexical Aspect: the case of semelfactives and degree adverbials. http://faculty.biu.ac.il/~rothss/rothstein%20-%20leipzig%20events%20paper.pdf

Schwarzschild, R. (2006). Dimensions in the syntax of nominal phrases. *Syntax* 9:1, 67−110;

Souckova, K. (2009). Kicking and slapping; On the event-internal/external distinction in Hausa. Handout "Nominal and Verbal Plurality", Paris, 7th November 2009. www.umr7023.cnrs.fr/sites/sfl/IMG/pdf/plnv09souckova.pdf

Zweig, E. (2009). Number-neutral bare plurals and the multiplicity implicature. *Linguistics and Philosophy*, 32:353–407.