Lars Hellan NTNU
Mary Esther Kropp Dakubu University of Ghana

Identifying Verb Constructions Cross-linguistically

May 21, 2009

List of contents

I. Introduction: a general overview of the system

We present here a system for sorting argument structures. Put another way, we introduce a system of labelling syntactic structures, in particular structures that have a verb as their head, according to their associated syntactic and semantic features. It has been developed by the authors in association with several others (see acknowledgements) within the typology program of the Legon-Trondheim Linguistics Project. Aspects of it have been presented in various fora from time to time during its development (eg. Dakubu 2008; Hellan 2008, 2009).

The system is aimed in the first place at facilitating comparative, typological research. To this end there has been every effort to make it notationally simple, to allow string-based search. For the same reason it is largely neutral as far as theoretical framework is concerned, and should be usable by linguists of all persuasions. The system provides a method for devising descriptive labels that include both syntactical and semantic information. Thus the system can help the typologist in elucidating how languages, whether they are related or not, express similar ideas syntactically, and whether apparently similar syntactic constructions are used for the same expressive purposes. The notion 'construction' is used in a theory neutral way, and refers to both the form and the content of an expression. What follows in this section is a brief overview of the system generally. Subsequent sections enumerate and explain the labels in more detail.

I.A. The notation

Technically, construction types are represented by strings of letters and hyphens called **templates**, composed by **labels**. Although some templates that involve relatively complex labels may seem daunting at first glance, the underlying principles are not complicated. We approach the construction from 'top' first noting its properties as a whole, and then properties of its main constituents, first their syntactic properties, then their semantic properties. This sequence is reflected in a notation from left to right. Each template thereby consists of several parts, referred to as **slots**.

A slot is filled by one or more labels expressing a clearly defined feature of the construction.
Slot 1 consists of a label for *Part of Speech* of the *head* of the entire construction, (in the system as developed so far, usually a verb) and—connected by underline—the category of possible *formatives* marked on the head. (The formatives may be realized as affixes, tones, stem formation (as in Semitic languages), vowel change, reduplication, and more – the realization mode as such is not displayed, only the category expressed.)

Slot 2 consists of a label for *valency specification* - like `intr` (intransitive), `tr` (transitive), `ditr` (ditransitive), and varieties thereof. This slot gives an over-all view of what kinds of arguments are expressed in the construction.

Slot 3 consists of one or more labels for specification of *syntactic constituents*: subject, object etc.

Slot 4 consists of one or more labels for specification of *participant roles*: agent, theme, instrument etc..

Slot 5 consists of a label for *aspect and Aktionsart*, written in CAPS.

Slot 6 consists of a label for the *situation type* or general semantics of the construction, written in CAPS.

Slot 7 provides a linking between the slot 6 situation type and the specifications in slots 2-4. This is of relevance especially for contents whose expression varies crosslinguistically (cf. (5)-(6) below), and for "idiomatic" or "metaphorical" constructions.

Of these, slots 1, 2 and 3 represent well understood areas of specification, and can build on much consensus across frameworks. Slots 4 and 5 are less robust, but have a core of consensus to build on. Slot 6 is still at a highly preliminary state of development. Slots 1 and 2 are obligatorily filled, the others not. A slot not filled is not displayed: the labels defined for the various slots are distinct and

quite unlike those for any other slot, hence no specification can be misread with regard to which slot it concerns. Likewise, no labels are distinguished in terms of CAP vs. not.

For the build-up of a template, the following conventions apply:

- Slots are interconnected by '-' (hyphen).
- Distinct items inside a slot are interconnected by '_' (underline).
- A label containing neither '-' nor '_' is an uninterrupted string of letters.
- If the content of a label is complex, the internal composition is indicated by alternation between small and capital letters (however, no labels are distinguished solely in terms of CAP vs. not).

We exemplify the notation with some templates. Constructions subsumed by the specification given in (1) are of a type one may expect to find in a very broad range of languages:

(1) `v-tr-suAg_obAffincrem-COMPLETED_MONODEVMNT`
(Ex.: English *the boy ate the cake*)

The template reads from left to right as follows:
Slot 1: the head is verb;
Slot 2: the syntactic frame is *transitive*;
Slot 4: the thematic roles expressed are *agent* `(ag)`, by Subject `(su)`, and *incrementally affected* `(affincrem)`, by Object `(ob)`;
Slot 5: the situation type is (partially) characterized as *completed monotonic development*.
Nothing occupies slots 3 and 6.

(2) and (3), exemplified from two languages spoken in Ghana, are also straightforward and widely attested, although the construction type in (3) is perhaps more localized:

(2) `v-intr-suAgmover-MOTION`
(Ex.: Ga *Kofi ba* 'Kofi came)

Slot 1 indicates that, like (1), the expression is headed by a verb, but Slot 2 indicates that unlike (1), the frame is intransitive. Slot 3 is again absent, but Slot 4 indicates that the role expressed by the Subject is an *agent mover* – a subtype of *agent*. Slot 5 is not present, but Slot 6 indicates that the situation type is characterized as MOTION.

(3) `v-tr-suAg_obThAbst-PROPTY`
(Ex.: Ewe É-wɔ akúvíá
3SG-do laziness 'He was lazy'

Reading the template from left to right, we find that in most respects the construction is identical to that of (1): Slot 1 indicates that the head is a verb; Slot 2 that the frame is transitive, Slot 4 that the subject expresses the role *agent*, and that the object expresses the role *abstract theme* – unlike the object of (1). Unlike (1) and (2), however, in this template Slot 6 is filled with the situation type PROPERTY, meaning that the whole expresses a property of the Subject, but Slot 5 is not filled.

Exemplifying with a construction type from Bantu languages illustrating *verbal extensions*; (4) is from Citumbuka (spoken in Northern Malawi and Zambia (courtesy of Jean Chavula)):

(4) `v-ditrOblApCs-oblCsu_obAobl-suCsr`
Tumbikani wa-ka-*mu*-phik-isk-ir-a *Temwa* nchunga kwa Mary
Tumbikani 1SM-pst-1OM-cook-Caus-Appl-fV Temwa beans 'to' Mary
'Tumbikani made Mary cook beans for Temwa'

The construction presents a person-causer and a three-participant caused event, with the Agent of the caused event (the 'Causee') expressed as Oblique, and an oblique participant of the caused event

having been promoted as an Applicative, taking the position of First Object. The component labels read as follows:

Valence slot (slot 2):
`ditrOblApCs`: double object plus oblique, built up—through 'backtracking' the operations—by 'Applicative Formation' and 'Causative Formation'.

Syntactic constituents' slot (slot 3):
`oblCsu`: the Oblique represents the 'causee', i.e., the subject relative to a 'base' structure composed by the same verb, , promoted by Causative Formation.
`obAobl`: the First Object represents an oblique relative to a 'base' structure composed by the same verb, promoted by Applative Formation ('`A`' for 'Applicative');

Semantic participants slot (slot 4):
`suCsr`: the Subject expresses a Causer

As is apparent from these examples, when comments are made about *constituents* of the construction, they are identified by the traditional *grammatical function* (GF) categories 'Subject', 'Object', 'Oblique', and the like – this applies whether the specifications are syntactic or semantic. In general there is only one of each GF per sentence. In cases where one speaks of a First Object and a Second Object, as in (4), these are counted as distinct GFs, whereas when a sentence has more than one Oblique, these will be distinguished Obl1, Obl2, etc. according to the order in which they occur. Apart from this reflection of linear order, the specification of constituents in a template says nothing about linear ordering.

The next example is again taken from Ga, illustrating a prevalent strategy in West African languages of using complex NPs and (di)transitivity for contents where e.g. English would use prepositions (Dakubu 2008):

(5) `v-ditr-obPostp-suAg_obEndpt_ob2Th-PLACEMENT`

Amɛ-wotsɔne	lɛ	mli	yɛlɛ
3P.AOR-put	vehicle	DEF	inside yam

'They put [vehicle's inside] [yam]' ='They put yams in the lorry.'

Here the two objects represent a Mover (the yam) as Second Object and its Endpoint (the lorry's inside) as First Object. No preposition exposes the Endpoint status. Moreover, this Endpoint is characterized as the inside of something else, but again no preposition is used, but rather something structurally like a possessive NP construction (often referred to as a 'postposition'). Thus, the labels read:

Valence slot:
`ditr`: double object construction;

Syntactic constituents slot:
`obPostp`: the First Object is a 'postpositional phrase', i.e., an NP with a head expressing a spatial domain relative to the item expressed in the Specifier of the NP;

Semantic participants slot:
`obEndpt`: the First Object represents the Endpoint of a movement;
`ob2Th`: the Second Object represents the Mover (Theme) of a movement;

Situation type slot:
`PLACEMENT`: The situation type is one of *placement* (putting something somewhere).

Another example from Ga (Dakubu op.cit.) exposes an *identity* (ID) and a *body-part* (BP) pattern:

(6) `v-tr-suPossp_obIDsuSpec-suBPsuSpec_suLocus_obExp-EXPER`

Mi-hiɛ di mi
`1S`$_1$`.POSS-face black 1S`$_1$
"My face blackens me" = 'I am dizzy.'

Syntactic constituents slot:
`suPossp`: the Subject is a possessive phrase (NP with an NP specifier)
`obIDsuSpec`: the Object is (referentially) **ID**entical to Specifier of the Subject
`suBPsuSpec`: the Subject is (referentially) a **B**ody**P**art of the Specifier of the Subject

Semantic participants slot:
`suLocus`: the subject expresses the 'locus' of the situation.
`obExp`: the Object expresses an Experiencer.

Situation type slot:
`EXPER`: The situation type is one of *experiencing* (someone having an experience).

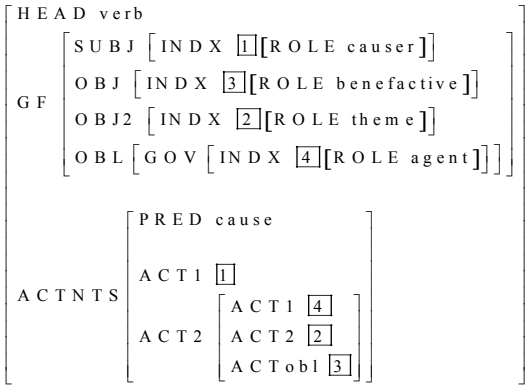
Below, in section II we present labels for slots 1, 2 and 3; in section III we present rather tentative labels for slots 4 and 5., and section IV presents template structures for various types of multi-verb constructions. As of now, the number of labels vailable in each slot domain is: Slot 2: 75; slot 3: 160; slot 4: 55; slot 5: 20, numbers based on development of the system for languages of the Volta Basin Area, some Bantu languages, and some Germanic languages especially Norwegian. A wiki page is currently in development at NTNU (Trondheim), where constructions and annotated example sentences can be viewed and discussed (www.typecraft.org). Thus, an inventory of Norwegian types is located at this site under www.typecraft.org/research/projects/Verbconstructions/ , and likewise one for Ga types.

Before displaying the labels, we show a linkage between the present code and a formalism used in some formal linguistic frameworks.

I.B. Linkage to AVM format

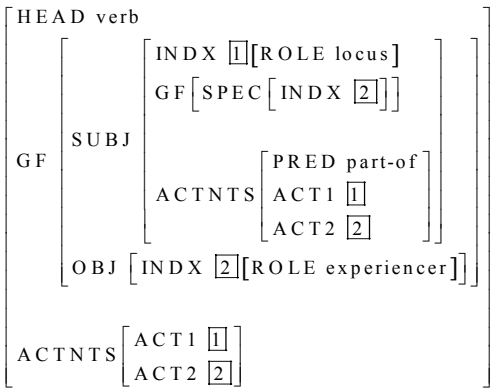
The template formalism is constructed in such a way as to be linkable to attribute-value-displays, as used for instance in HPSG and LFG. For instance, the information encoded in the template (4) above can be exposed in AVM (Attribute Value Matrix) notation as shown in (7), with `GF` standing for 'grammatical functions', `ACTNTS` for 'actants' (= 'participants'), and `ACTn` used according to the convention that given the situation type expressed by the verb, a participant with the role carried by `ACTn+1` could not be expressed unless the role carried by `ACTn` is expressed. These labels are a blend from many frameworks, such as GF-notions from LFG, semantic notions from Melchuk, integrated syntactic-semantic description partly in the spirit of Melchuk and HPSG. (Left out in (7) is an exposition of possible intermediate steps of the *derivational* processes 'Applicative' and 'Causative' – the `ACTNTS` structure here mirrors a possible 'base' configuration, and `GF` exposes the resulting syntactic functional structure.)

(7)



An AVM corresponding to the specification of (6) will be as in (8), using the same design as in (7):

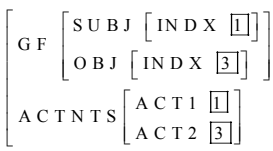
(8)



From a technical point of view, it is possible to model each separate label as a partial AVM, so that, with '-' and '_' in the templates interpreted as *unification* operators, the AVMs of labels constituting a template can be merged together to an AVM of the entire template. With the template in (6) and the AVM in (8) as an example, the constituent labels of (6) can be defined as the AVMs listed in (9); merging them yields (8):

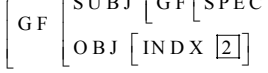
(9)

`tr` =>

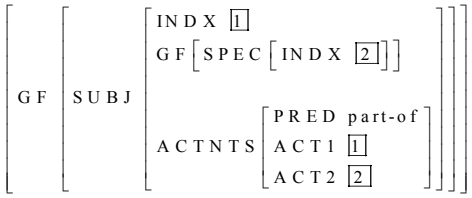


`suPossp` => [GF [SUBJ [GF[SPEC]]]]

`obIDsuSpec` =>



`suBPsuSpec` =>



`suLocus` => [GF [SUBJ [INDX [ROLE locus]]]]

`obExp` => [GF [OBJ [INDX [ROLE experiencer]]]]

In the definitions in section II below, all labels are associated with such AVMs, along with definitions in words spelling out the intended content. For any combination of labels constituting a template, such a merged AVM can be constructed.

To indicate the space of specifications considered, Table 1 gives a list of attributes serving inside of the AVMs. In this list, features in boldface are 'outermost' in a sign path, and features in italics are next in the path. In addition to explaining the contents of the AVMs, this list also summarizes most of the factors of verb constructions that the system currently addresses.

Table 1
Attributes and Values Employed

HEAD	part of speech and other properties associated with the head of a construction
FORMATIVES	list of affixes, tones, stem formation (as in Semitic), reduplication, and other formatives marked on the head constituent
CASE	case (mainly for nouns, pronouns and determiners)
DEF	definiteness (mainly for nouns, pronouns and determiners)
REAL	realization status: dropped, cliticized, normal (mainly for pronouns)
AGR-TARGET	the constituent is targeted by agreement marking on the head of the construction (mainly for nominals)
TAM	Tense/aspect/mood (mainly for verbs)
GF	grammatical function
SUBJ	subject sign
OBJ	object sign; used together with IOBJ, OBJ is 'direct object', and together with OBJ2, 'first object'
IOBJ	indirect object, to be used in combination with OBJ
OBJ2	second object, to be used in combination with OBJ
COMP	sentential complement (not being classified as object)
OBL	oblique, i.e., a PP where the governed NP has a role defined relative to the head, and it thus is the semantics of that NP, and not the semantics of the PP as a whole, which is of interest
PRESENTED	'presented' NP in a presentational construction
SECPRD	secondary predicate
IDNT	complement of an identifying predicate
ADVBL	'adverbial complement', i.e., a PP, Adv or AdvP serving as complement, where it is the semantics of the whole constituent which is of interest
PRTCL	'particle', with aspectual or less tangible impact
VID	'verbid', a VP serving a bit like an OBL
GOV	governee, used in connection with a preposition for its inherent GF (roughly, an abbr. for 'GF | OBJ')
INDX	referential index
ROLE	participant role ('theta-role')
CLASS	class, i.e., inherent properties
XACT	'exposed actant': in 'raising' and 'equi' constructions, XACT coincides with the subject of the infinitive, and in non-verbal secondary predicates it coincides with the ACT1 of the predicate.
ACTNTS	'actants', i.e., participants of the situation type expressed by the head of the construction

ACT0	index of the situation type expressed by the construction
ACT1	actant 1
ACT2	actant 2
ACT3	actant 3
ACTobl	actant expressed by the NP complement of an oblique
LOC	locative argument
DIR	directional argument
PRED	predicate (used only with grammatically expressed meanings)
ASPECT	aspect
AKTART	Aktionsart

Values

+/-	
copula	value of HEAD: a subtype of *verb*
drop	value of HEAD | REAL: dropped, in the sense 'pro-drop'
clit	value of HEAD | REAL: cliticized
nomin	value of HEAD | CASE
acc	value of HEAD | CASE
dat	value of HEAD | CASE
gen	value of HEAD | CASE
abl	value of HEAD | CASE
ill	value of HEAD | CASE
abs	value of HEAD | CASE
erg	value of HEAD | CASE
decl-compl	value of HEAD
yes-no-compl	value of HEAD
wh-compl	value of HEAD
infin-compl	value of HEAD
gerund	value of HEAD | TAM
infinitive	value of HEAD | TAM
irrealis	value of HEAD | TAM
cause	value of ACTNTS| PRED
increm-cause	value of ACTNTS| PRED (causation happening incrementally)
binary-rel	value of ACTNTS| PRED
part-of	value of ACTNTS| PRED
spatial-coord-of	value of ACTNTS| PRED
concur	value of ACTNTS| PRED
explet	value of INDX: expletive, i.e., referentially void
spatial	value of INDX | CLASS
bodypart	value of INDX | CLASS
sign	value of any GF|SUBJ, GF|OBJ, GF|IOBJ, etc.: sign
oriented-obj	value of ACT1 and ACT2: oriented object, a super-type of paths, direction indicators and locomotors (movers)

Of the attributes in Table 1, the GF attributes correspond to the initial part of any Slot 3 or Slot 4 label, abbreviated as follows:

(10)

SUBJ	su
OBJ	ob
IOBJ	iob
OBJ2	ob2
COMP	comp
OBL	obl
PRESENTED	pres
SECPRD	sc
IDNT	idnt
ADVBL	adv
PRTCL	prtcl
VID	vid

What follows the initial part `su`, `ob`, etc. *may* correspond to an attribute path in an AVM leading 'in' from 'GF', but may equally well skip to some item deeper into the path, as when the label `suClit` corresponds to 'SUBJ | HEAD | REAL clit'.

Exactly which of the items in (10) appear in an AVM depends on the label occupying Slot 2: with `intr,` only SUBJ occurs, with `tr,` both SUBJ and OBJ occur, with `ditr,` both SUBJ, OBJ and IOBJ occur. If ..`Obl` occurs in the Slot 2 label, then OBL occurs in the AVM; and as for the other attributes in (10), their 'licensing' Slot 2 counterparts can be inferred from the labels overview in section II.

The valence-labels and concepts `intr`, `tr`, and `ditr` are based on the following definitions:

A ***direct syntactic argument*** of a verb is any nominal constituent syntactically directly related to the verb (as subject-of, direct object-of, or indirect object-of), and any clausal constituent with either of these functions. This *in*cludes expletive subjects and objects, and *ex*cludes clausal constituents in extraposed position; it also excludes any NP or clause governed by a preposition (thus, any `obl`). It also excludes NPs carrying locative case as in Finno-Ugric or Caucasian languages – these count as obliques – see below.

With this notion of 'direct syntactic argument', we define the three basic valency notions:

`intr` = intransitive, i.e., with only SUBJECT as direct syntactic argument.

`tr` = transitive, i.e., with SUBJECT and one OBJECT as direct syntactic arguments.

`ditr` = ditransitive, i.e., with SUBJECT and two OBJECTs as direct syntactic arguments. (Also: '`dbob`' = 'double object' is used, with the same definition.)

A direct syntactic argument is ***standardly linked*** when it has referential content and serves a semantic argument function relative to the verb. (This *ex*cludes expletive subjects and expletive objects, and 'raised' full NPs.) Linking is reflected in the AVM in the identities between specifications under GF and under ACTNTS.

Any standard introduction to feature structure notation, or to LFG or HPSG, gives an introduction to AVM notation. For the particular feature geometry used here, an introduction is given in Hellan 2009, and for a toolkit for building elementary grammar fragments using this AVM notation, see Hellan 2008b.

II. Labels for slots 1, 2 and 3 in Single-verb constructions

II.A. SLOT 1 Head specification

There are many possible combinations of formatives. Only a few are entered here in Table 2.

Table 2.
Sample Labels for Slot 1

v = construction is headed by Verb.
[HEAD verb]

v_pas = construction is headed by Verb and the verb has a Passive formative
[HEAD verb[FORMATIVES ⟨passive⟩]]

v_prf = construction is headed by Verb and the verb has a Perfect formative

v_aor = construction is headed by Verb and the verb has an Aorist formative

v_prog = construction is headed by Verb and the verb has a Progressive formative

v_hab = construction is headed by Verb and the verb has a Habituial formative

v_sm = construction is headed by Verb and the verb has a Subject Marker formative. This and several following are used for languages where arguments must be marked on the verb according to syntactic function.

v_om = construction is headed by Verb and the verb has an Object Marker formative

v_agr = construction is headed by Verb and the verb has an Agreement formative (used only for languages/constructions where there is no contrast between Subject Marker and Object Marker)
[HEAD verb[FORMATIVES ⟨AGR⟩]]

v_smOm = construction is headed by Verb and the verb has a Subject Marker and an Object Marker formative
[HEAD verb[FORMATIVES ⟨SM, OM⟩]]

v_appl = construction is headed by Verb and the verb has an Applicative formative

v_applPas = construction is headed by Verb and the verb has an Applicative and a Passive formative

v_caus = construction is headed by Verb and the verb has a Causative formative

v_causPas = construction is headed by Verb and the verb has a Causative and a Passive formative

v_causAppl = construction is headed by Verb and the verb has a Causative and an Applicative formative

v_causApplPas = construction is headed by Verb and the verb has a Causative, an Applicative and a Passive formative

v_causSmOm = construction is headed by Verb and the verb has a Causative formative, a Subject Marker and an Object Marker
[HEAD verb[FORMATIVES ⟨causative, SM, OM⟩]]

II.B. SLOT 2 Valence (see end of this section for *derived* valence)

The following general definitions are essential to Slot 2-definitions (restating from the end of I.b):

A ***direct syntactic argument*** of a verb is any nominal constituent syntactically directly related to the verb (as subject-of, direct object-of, or indirect object-of), and any clausal constituent with either of these functions. This *in*cludes expletive subjects and objects, and *ex*cludes clausal constituents in extraposed position; it also excludes any NP or clause governed by a preposition. It also excludes NPs carrying locative case as in Finno-Ugric or Caucasian languages – these count as obliques – see below.

With this notion 'direct syntactic argument', we define three basic valency notions:

intr = intransitive, i.e., with only SUBJECT as direct syntactic argument.
tr = transitive, i.e., with SUBJECT and one OBJECT as direct syntactic arguments.
ditr = ditransitive, i.e., with SUBJECT and two OBJECTs as direct syntactic arguments. (Also: 'dbob' = 'double object' (same definition).)

A direct syntactic argument is ***standardly linked*** when it has referential content and serves a semantic argument function relative to the verb. (This *ex*cludes expletive subjects and expletive objects, and 'raised' full NPs.)

The following list contains all defined Slot 2 labels.

intr = intransitive, i.e., with only SUBJECT as direct syntactic argument, standardly linked.

$$\begin{bmatrix} \text{GF} \left[\text{SUBJ} \left[\text{INDX } \boxed{1}\right]\right] \\ \text{ACTNTS} \left[\text{ACT1 } \boxed{1}\right] \end{bmatrix}$$

(Ex.: Eng. *he sleeps*)

intrImpers = impersonal intransitive, i.e., SUBJECT is an expletive not linked to any other item in the clause.

$$\begin{bmatrix} \text{GF} \left[\text{SUBJ} \left[\text{INDX expl}\text{et}\right]\right] \\ \text{ACTNTS} \left[\ \right] \end{bmatrix}$$

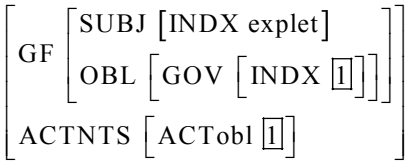
(Ex.: Eng. *it snows*)

intrImpersPrtcl = impersonal intransitive with an aspectual particle.

$$\begin{bmatrix} \text{GF} \begin{bmatrix} \text{SUBJ} \left[\text{INDX expl}\text{et}\right] \\ \text{PRTCL sign} \end{bmatrix} \\ \\ \text{ASPECT aspect} \\ \text{ACTNTS} \left[\ \right] \end{bmatrix}$$

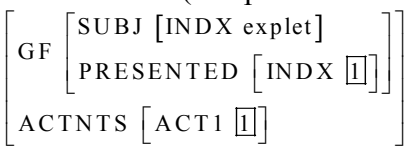
(Ex.: Norw. *det klarner opp* 'it clears up')

intrImpersObl = impersonal intransitive with an Oblique argument.



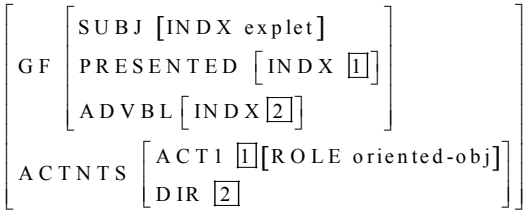
(Ex.:Norw. *det synger i fjellene* 'it sings in the mountains'
= 'one can hear singing from inside of the mountains')

intrPresnt = intransitive presentational, i.e., an expletive subject and an indefinite NP (the 'presented' NP) occupying the post-verbal position.



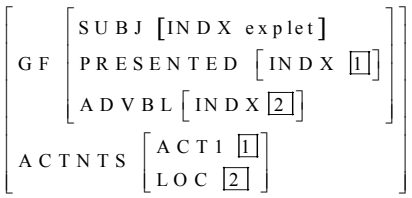
(Ex.: Eng. *there lives a man*)

intrPresntPath = intransitive presentational with a Path adverbial**.**



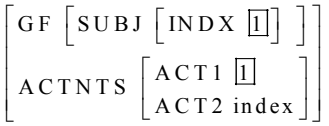
(Ex.: Norw. *det springer en mann nedover bakken*
'there runs a man down the hillside')

intrPresntLoc = intransitive presentational with a Locative adverbial.



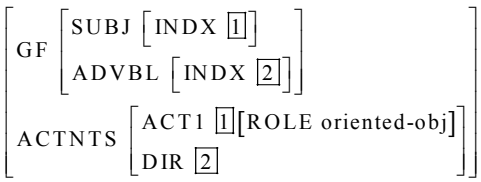
(Ex.: Norw. *det sitter en mann i stolen*
'there sits a man in the chair')

intrImplobj = intransitive with an implicit object.



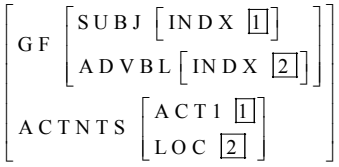
(Ex.: Eng. *he ate*)

intrPath = intransitive with a Path adverbial.



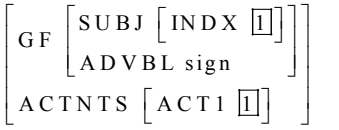
(Ex.: Eng. *he drove to Finnmark*)

intrLoc = intransitive with a ('bound') locative adverbial.



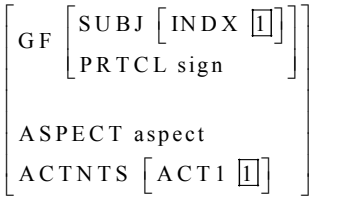
(Ex.: Eng. *he lives in Finnmark*)

intrAdv = intransitive with a ('bound') Manner adverbial.



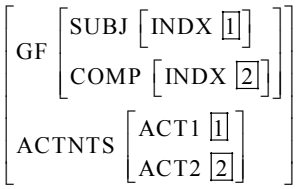
(Ex.: Eng. *he functions well*)

intrPrtcl = intransitive with an aspectual particle.



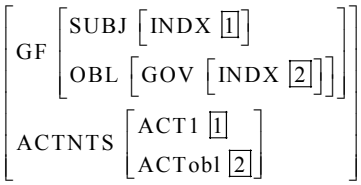
(Ex.: Norw. *regnet varer ved* 'the rain lasts')

intrComp = intransitive with a sentential complement (not classifiable as object).



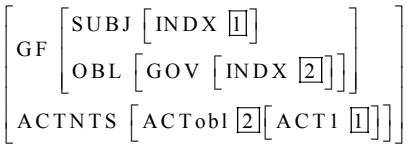
(Ex.: Ga *Yoo lɛ e-tee ní e-ya-he wolo lɛ*
woman$_1$ DEF PERF-go COMP 3S$_1$-EGR-buy book DEF
'The woman has gone to buy a book')

intrObl = intransitive with an Oblique (PP) argument.



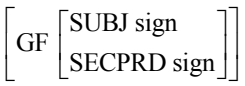
(Ex.: Engl. *he talks about John*)

intrOblRais = intransitive with an oblique argument from which an NP has been 'raised'.



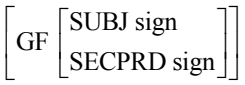
(Ex.: Norw. *han later til å komme* 'he appears [to] to come')

intrScpr = intransitive with a secondary predicate ('Small Clause' predicate).



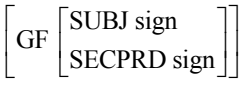
(Ex.: Eng. *he seems sick*)
(For further classification, see Slot 3, labels starting with **sc**...)

intrLghtScpr = intransitive light verb with a secondary predicate (see near-equivalents **lghtAdj/ lghtAdv/ lghtN** below).



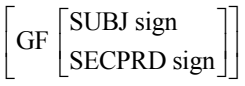
(Ex.: Eng. *the house stands empty*)

intrAuxperfScpr = intransitive perfect auxiliary verb with a secondary predicate. [This presupposes a 'raising analysis of auxiliaries. See 'axv' in the Multiverb section.]



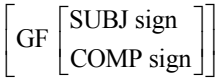
(Ex.: *he has arrived*)

intrAuxmodScpr = intransitive modal auxiliary verb with a secondary predicate ('epistemic modal'). [This presupposes a 'raising analysis of auxiliaries. See 'axv' in the Multiverb section.]



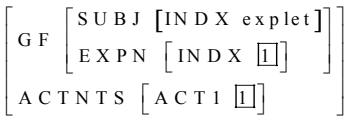
(Ex.: *he will arrive*)

intrAuxmodComp = intransitive modal auxiliary verb with a complement. ('root modal') [This presupposes a 'raising analysis of auxiliaries. See 'axv' in the Multiverb section.]



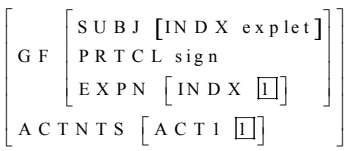
(Ex.: Eng. *he can sing*)

intrExpn = intransitive with an 'extraposed' clause.



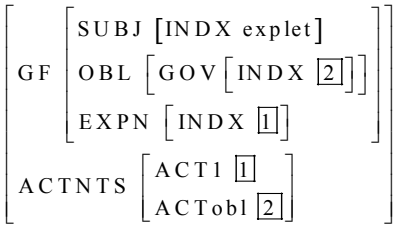
(Ex.: Eng. *it seems that he is sick*)

intrPrtclExpn = intransitive with an 'extraposed' clause and adverbial particle.



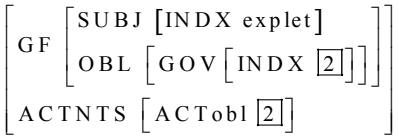
(Ex.: Eng. *it came out that he was sick*)

intrOblExpn = intransitive with an 'extraposed' clause and an oblique argument.



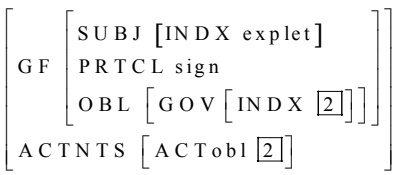
(Ex. Eng. *It depends on you whether he will win*)

intrOblExlnk = intransitive with an 'extralinked' clause and an oblique argument. [An *extralinked* clause is like an *extraposed* clause except that substituting it for the expletive does not yield a grammatical construction.]



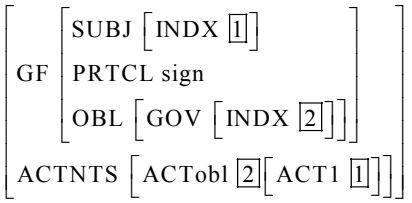
(Ex.:Norw. *det haster med å rydde*
'it hastes with to tidy' = "it is urgent that it gets tidied up")

intrPrtclOblExlnk = intransitive with an 'extralinked' clause, an oblique argument, and an advparticle. [An *extralinked* clause is like an *extraposed* clause except that substituting it for the expletive does not yield a grammatical construction.]



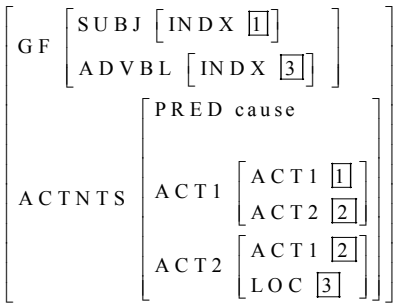
(Ex.: Norw. *det ser ut til at han kommer*
'it looks out to that he comes' = "it seems that he comes")

intrPrtclOblRais = intransitive with an oblique argument from which an NP has been 'raised', and an adverbial particle.



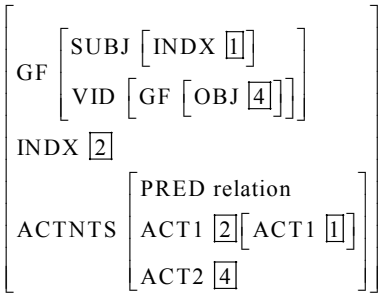
(Ex.:Norw. *han ser ut til å komme*
'he looks out to to come' = "he seems to come")

intrImpltransfAdv = intransitive with adverbial, with an implicit object transferred.



(Ex.: Eng. *he vomited on himself*)

intrVid = intransitive together with a verbid phrase[1]



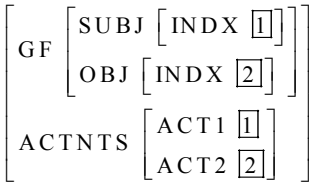
(Ex. Ga: E-da fe mi
3S-grow surpass 1S
'He is bigger than me.')

intrVidScpr = intransitive with a secondary predicate and a verbid phrase

intrSubcoord = intransitive with a subcoordination
(Ex. Norw.: Ola driver og plystrer
'Ola keeps and whistles' = "Ola keeps whistling"

intrPrtclSubcoord = intransitive with a particle and a subcoordination
(Ex. Norw.: Ola driver på og plystrer
'Ola keeps on and whistles' = "Ola keeps on whistling"

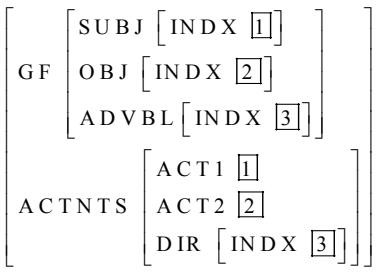
tr = transitive, i.e., with SUBJECT and one OBJECT, standardly linked.



(Ex.: Eng. *he kicked the ball*)

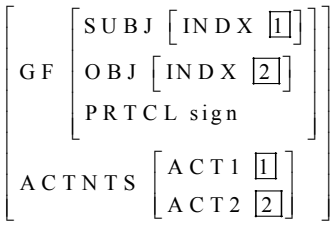
[1] For a discussion of verbid expressions as they appear in Ga see Dakubu xx

trPath = transitive, where the subject or object is understood in a directional capacity, and a path specification.



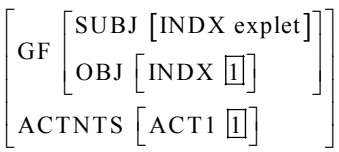
(Ex.: Eng. Directional subj: *he passed a church along the road*
Directional obj: *he threw the ball through the window*)

trPrtcl = transitive with an adverbial particle.



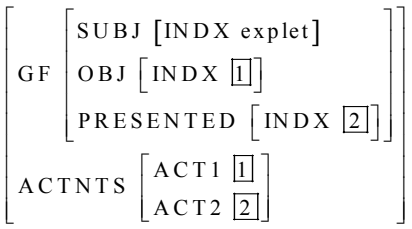
(Ex.: Norw. *Kari fant ut svaret* 'Kari found out the answer')

trImpers = impersonal transitive, where SUBJECT is an expletive not linked to any other item in the clause.



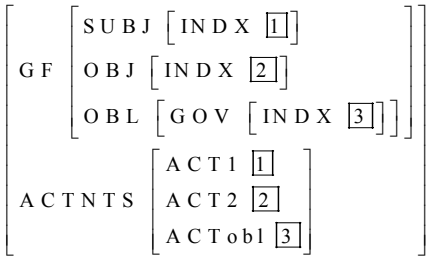
(Ex.: Ga *E-fi mi*
3S-tie 1S
'I am in difficulties.')

trPresnt = presentational with an NP (object) preceding the 'presented' NP.



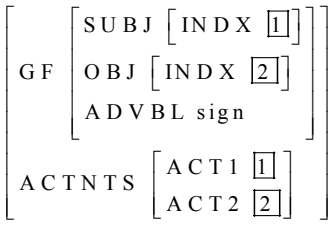
(Ex.: Norw. *det venter ham en ulykke*
'there awaits him an accident' = "an accident awaits him".)

trObl = transitive with an oblique.

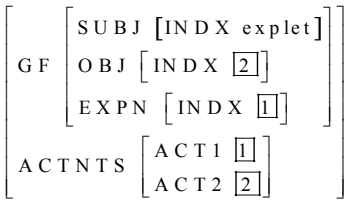


(Ex.: Eng. *he told Peter about the window*)

trAdv = transitive with an obligatory adverbial.

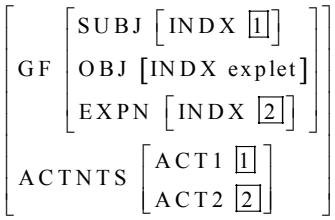


trExpnSu = transitive with an extraposed clause correlated with the subject, and an argument object.



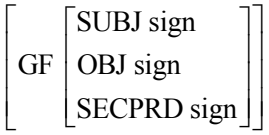
(Ex.: Eng. *it impresses me that he can sing*)

trExpnOb = transitive with an extraposed clause correlated with the object, and an argument subject..



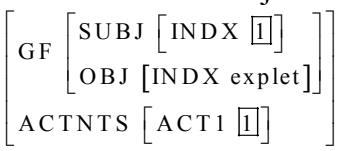
(Ex.: Norw. *vi muliggjorde det at han fikk innreisetillatelse*
'we possible-made it that he got entrance visa'
=. "we made it possible for him to get an entrance visa")

trScpr = transitive with a secondary predicate ('Small Clause' predicate).



(Ex.: Eng. *he made me sick*)
(For further classification, see slot 3, with **sc**...)

trNrf = transitive whose object is non-referential.



(Ex.: Norw. *Kari skammer seg*
'Kari shames herself' = "Kari is ashamed")

trNrfScpr = transitive whose object is non-referential, and with a secondary predicate

(Ex.: Norw. han viser seg å komme
'he shows REFL to come' = "he turns out to come")

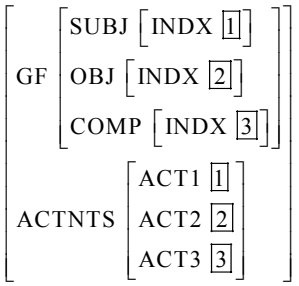
trNrfExpnSu = transitive whose object is non-referential, and with an 'extraposed' clause linked to subject.

(Ex. Norw: det viser seg at han kommer
'it shows itself that he comes' = "it turns out that he comes")

trNrfPresntLoc = transitive presentational with a non-referential object, and with a locative

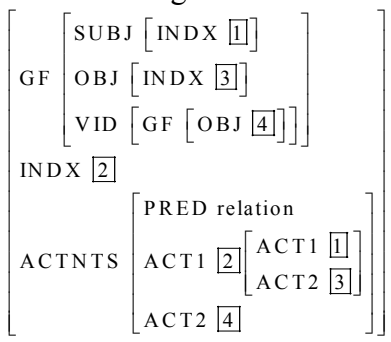
(Ex. Norw.: det oppholder seg en gutt i hagen 'there stays REFL a boy in the garden' = "there is a boy staying in the garden")

trComp = transitive with a sentential complement (apart from the object).



(Ex.: Ga: Ò-bàá-nyɛ́ éné ó-lá?
2S-INGR.FUT-able this 2S.SBJV-sing
Can you sing this? (are you capable of this that you could sing it?)

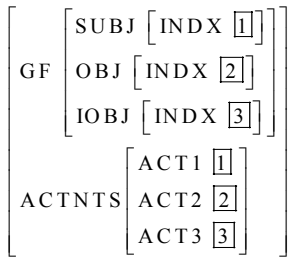
trVid = transitive together with a verbid phrase



(Ex.: Ga: E-ye loo fe mi
3S-eat meat surpass 1S
'She ate more meat than me.')

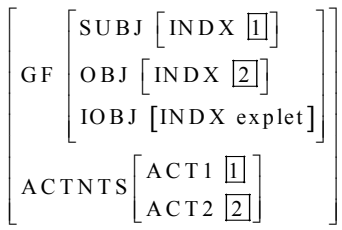
trLghtVid = transitive light verb with a verbid

ditr = ditransitive, i.e., with SUBJECT and two OBJECTs (here referred to by the traditional terms 'indirect' ('iob') and 'direct' object), standardly linked.



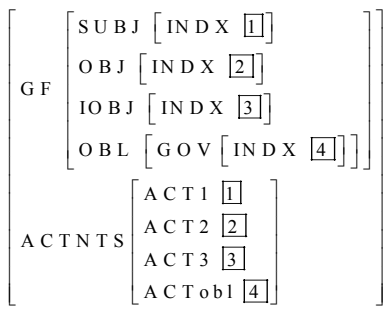
(Ex.: Eng. *he gave me the book*)

ditrNrf = ditransitive whose indirect object is non-referential.



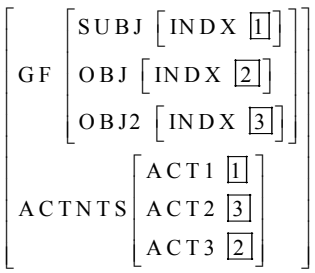
(Ex.: Norw. *han foresetter seg å komme*
he [foresetter] himself to come' = "he plans on coming")

ditrObl = ditransitive with oblique.



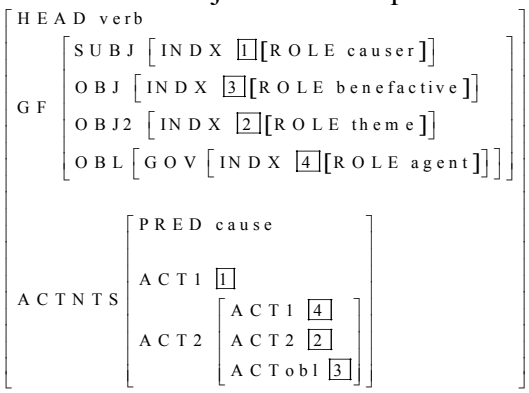
(Ex.: Norw, *jeg kaster Ola kakestykker i ansiktet*
'I throw Ola cakes in the face' = "I throw cakes in the face of Ola")

dbob = double object, i.e., with SUBJECT and two OBJECTs referred to by the terms '(first) object' and 'second object'), standardly linked.



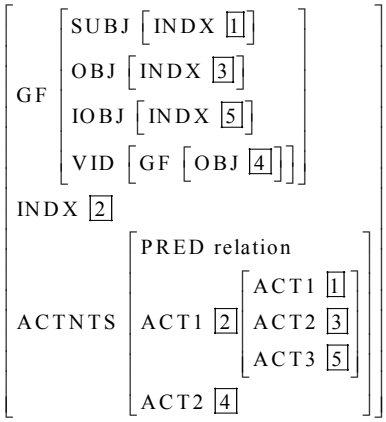
(Ex.Citumbuka:
Tumbikani wa-ka-*mu*-pa *Mary* ndalama
Tumbikani 1SM-pst-1OM-give Mary money
'Tumbikani gave Mary money.')

dbobObl... = double object with oblique.



(Ex.Citumbuka (really `ditrOblApCs` – see Introduction, (4)):
Tumbikani wa-ka-*mu*-phik-isk-ir-a *Temwa* nchunga kwa Mary
Tumbikani 1SM-pst-1OM-cook-Caus-Appl-fV Temwa beans 'to' Mary
'Tumbikani made Mary cook beans for Temwa')

ditrVid = ditransitive together with a verbid

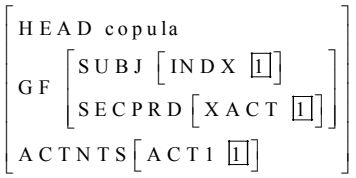


(Ex.Ga:

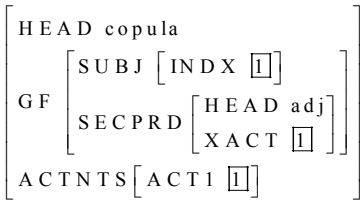
wɔ-bi	Ataa Kwami	shi	yɛ	e-dɛ-ŋ
1P-ask	A.K.	down	be.at	3SPOSS-hand-LOC
V	N	N	V	NP

'We asked him about Ataa Kwami')

predicative copular construction = construction where the verb ties an NP and a predicate together so as to make the NP the logical subject (XACT) of the predicate



copAdj = *predicative copular construction* with adjectival predicative.



copN = *predicative copular construction* with nominal predicative.

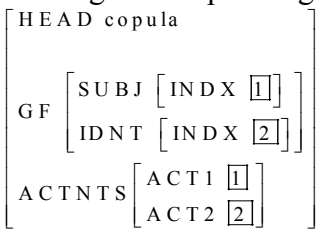
copPP = *predicative copular construction* with prepositional predicative.

copPredprtcl = *predicative copular construction* with predicative headed by a predicative particle.

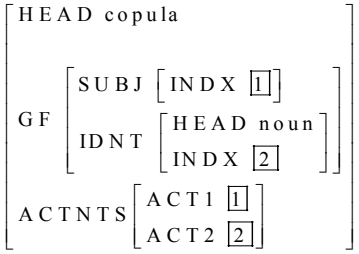
coplocAdj = *predicative copular construction* with adjectival predicative and where the verb (like *yɛ* 'be.at' in Ga) suggests the predicate as somehow a location.

coplocAdv = *predicative copular construction* with adverbial predicative and where the verb (like *yɛ* 'be.at' in Ga) suggests the predicate as somehow a location.

identity copular construction = construction where the verb ties two referring expressions together expressing identity between their referents



copIdN = *identity copular construction* with nominal identifier.



(Ex.: Norw. *dette er mannen*
'this is the man'.)

copIdAbsinf = *identity copular construction* with infinitival identifier.

(Ex.: Norw. *oppgaven er å spise silden*
'the task is to eat the herring'.)

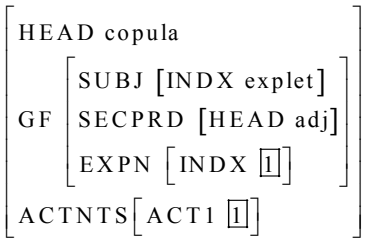
copIdDECL = *identity copular construction* with a declarative clause as identifier.

(Ex.: Norw. *problemet er at han spiser silden*
'the problem is that he eats the herring'.)

copIdYN = *identity copular construction* with a yes-no-interrogative clause as identifier.
(Ex.: Norw. *problemet er om han spiser silden*
'the problem is whether he eats the herring'.)

copIdWH = *identity copular construction* with a wh-interrogative clause as identifier.
(Ex.: Norw. *spørsmålet er hvem som spiser silden*
'the question is who eats the herring'.)

copExpnAdj = *predicative copular construction* with adjectival predicative and the 'logical subject' extraposed.



(Ex.: Norw. *det er trist at han kommer*
'it is sad that he comes';
det er uvisst hvem som kommer
'it is uncertain who comes'.)

copExpnN = *predicative copular construction* with nominal predicative and the 'logical subject' extraposed.
(Ex.: Norw. *det er en skuffelse at han kommer*
'it is a disappointment that he comes';
det er et spørsmål hvem som kommer
'it is a question who [that] comes'.)

copExpnPP = *predicative copular construction* with prepositional predicative and the 'logical subject' extraposed.
(Ex.: Norw. *det er hinsides diskusjon at han kommer*
'it is beyond discussion that he comes'.)

copExpnPredprtcl = *predicative copular construction* with predicative headed by a pred-particle and the 'logical subject' extraposed.
(Ex.: Norw. *det var som bestilt at han tapte igjen*
'it was like booked that he lost again'.
= "it was as one would have wished that he lost again")

lghtAdj = intransitive light verb whose complement is headed by Adj functioning as a secondary predicate (= **intrLghtScpr-scAdj** – see above).

lghtAdv = intransitive light verb whose complement is headed by Adv functioning as a secondary predicate (= **intrLghtScpr-scAdv** – see above).

lghtN = intransitive light verb whose complement is N functioning as a secondary predicate (= **intrLghtScpr-scN** – see above)
(in contrast to **trLght** – see below).

lghtAdjVid = intransitive light verb whose complement is headed by Adj functioning as a secondary predicate, and with a Verbid phrase

trLght = transitive light verb whose complement is an NP expressing an event-type performed (or in other ways operated on) by the subject.

(Ex. Eng.: he makes progress.)

Derivational (Operational) history

Below are labels reflecting derivational/ operational history (like Passive, Applicative, Causative, etc.). In the explanation, '>' means "applying before". The labels 'unwrap' the derivational history, starting with a symbol for the actual valence, then a symbol for the 'last' derivational process leading up to this valence, then the 'second last' derivational process, and so forth. See **section V** for a discussion, and overview of the component parts.

intrPs = intransitive resulting from Passive; root transitive
intrPsAp = intransitive resulting from Passive following Applicative (A>P; root intransitive)
intrPsCs = intransitive resulting from Passive following Causativization (C>P; root intransitive)
intrRf = intransitive resulting from Reflexivization; root transitive
intrRp = intransitive resulting from Reciprocization; root transitive
intrSt = intransitive resulting from Stativization; root transitive
intrOblPsCs = intransitive oblique resulting from Passive following Causativization (C>P; root intransitive)
trAp = transitive resulting from Applicative; root intransitive
trCs = transitive resulting from Causativization; root intransitive
trApCs = transitive resulting from Applicative following Causativization (C>A; root intransitive)
trPsAp = transitive resulting from Passive following Applicative (A>P; root transitive)
trPsCs = transitive resulting from Passive following Causativization (C>P; root transitive)
trPsApCs = transitive resulting from Passive following Applicative following Causation (C>A>P; root intransitive)
trRf = transitive resulting from Reflexivization; root ditransitive
trRfAp = transitive resulting from Reflexivization following Applicative (A>Rf; root transitive)
trRfApCs = transitive resulting from Reflexivization following Applicative following Causation (C>A>Rf; root intransitive)
trRp = transitive resulting from Reciprocization; root ditransitive
trRpAp = transitive resulting from Reciprocization following Applicative (A>Rp; root transitive)
trRpApCs = transitive resulting from Reciprocization following Applicative following Causation (C>A>Rp; root intransitive)
trOblCs = transitive oblique resulting from Causativization; root transitive
ditrAp = ditransitive resulting from Applicative; root transitive
ditrCs = ditransitive resulting from Causativization; root transitive
ditrPsCs = ditransitive resulting from Passive following Causativization (C>P; root ditransitive)

ditrPsApCs = ditransitive resulting from Passive following Applicative following Causation (C>A>P; root transitive)
ditrOblCs = ditransitive oblique resulting from Causativization; root ditransitive
ditrOblApCs = ditransitive resulting from Applicative following Causativization (C>A; root transitive)
tritrAp = tritransitive resulting from Applicative; root ditransitive
tritrCs = tritransitive resulting from Causativization; root ditransitive
tritrApCs = tritransitive resulting from Applicative following Causativizaton (C>A; root transitive)
tritrPsCs = tritransitive resulting from Passive following Causativization (C>P; root ditransitive)
tritrPsApCs = tritransitive resulting from Passive following Applicative following Causativization (C>A>P; root ditransitive)
qtrApCs = quatrotransitive resulting from Applicative following Causativization (C>A; root ditransitive)
dbobAp = **ditrAp** = double-object resulting from Applicative; root transitive
dbobCs = **ditrCs** = double-object resulting from Causativization; root transitive
dbobPsCs = **ditrPsCs** = double-object resulting from Passive following Causativization (C>P; root ditransitive)
dbobPsApCs = **ditrPsApCs** = double-object resulting from Passive following Applicative following Causation (C>A>P; root transitive)
dbobOblCs = **ditrOblCs** = double-object oblique resulting from Causativization; root ditransitive
dbobOblApCs = **ditrOblApCs** = double-object resulting from Applicative following Causativization (C>A; root transitive)
triobAp = **tritrAp** = triple-object resulting from Applicative; root ditransitive
triobCs = **tritrCs** = triple-object resulting from Causativization; root ditransitive
triobApCs = **tritrApCs** = triple-object resulting from Applicative following Causativizaton (C>A; root transitive)
triobPsCs = **tritrPsCs** = triple-object resulting from Passive following Causativization (C>P; root ditransitive)
triobPsApCs = **tritrPsApCs** = triple-object resulting from Passive following Applicative following Causativization (C>A>P; root ditransitive)
qtrobApCs = **qtrApCs** = quadruple-object resulting from Applicative following Causativization (C>A; root ditransitive)

II.C SLOT 3 Constituents, syntactic properties

(see end of this section for *derived* GFs)

suExpl = subject is an expletive.

$$\left[\text{GF}\left[\text{SUBJ}\begin{bmatrix}\text{HEAD pron}\\ \text{INDX explet}\end{bmatrix}\right]\right]$$

suDir = object is understood in a directional capacity.

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{INDX}\left[\text{ROLE oriented-obj}\right]\right]\right]\right]$$

suDECL = subject is a declarative clause.

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD decl-comp}\right]\right]\right]$$

suYN = subject is a yes-no-interrogative clause.

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD yes-no-comp}\right]\right]\right]$$

suWH = subject is a wh-interrogative clause.

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD wh-comp}\right]\right]\right]$$

suInf = subject is an infinitival clause.

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD infin-comp}\right]\right]\right]$$

suGer = subject is a gerundive clause.

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD verb}\left[\text{TAM gerund}\right]\right]\right]\right]$$

suAbsinf = subject is an infinitival clause with non-controlled interpretation.

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD infin-comp}\right]\right]\right]$$

suNrg = subject is a non-argument.

suUnif = subject unifies with the verb to determine the verbal meaning

suSM = subject is targeted by the verb's subject marking

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{AGR-TARGET +}\right]\right]\right]\right]$$

suAgr = subject is targeted by the main verb's agreement marking

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{AGR-TARGET +}\right]\right]\right]\right]$$

suAgraux = subject is targeted by the auxiliary verb's agreement marking

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{AGR-TARGET +}\right]\right]\right]\right]$$

suAgrsc = subject is targeted by the secondary predicate's agreement marking

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{AGR-TARGET +}\right]\right]\right]\right]$$

suNom = subject has case Nominative

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{CASE nomin}\right]\right]\right]\right]$$

suAcc = subject has case Accusative

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{CASE acc}\right]\right]\right]\right]$$

suGen = subject has case Genitive

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{CASE gen}\right]\right]\right]\right]$$

suDat = subject has case Dative

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{CASE dat}\right]\right]\right]\right]$$

suErg = subject has case Ergative

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{CASE erg}\right]\right]\right]\right]$$

suAbsl= subject has case Absolutive

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD}\left[\text{CASE absol}\right]\right]\right]\right]$$

suClit = subject is cliticized (cliticization site not specified)

$$\left[\text{GF}\left[\text{SUBJ}\left[\text{HEAD pron}\left[\text{REAL clit}\right]\right]\right]\right]$$

suObClit = subject and object are cliticized (cliticization site not specified)

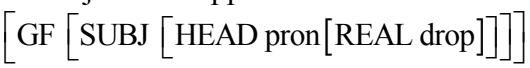
suObIobClit = subject and object and indirect object are cliticized (cliticization sites not specified)

suIobClit = subject and indirect object are cliticized (cliticization sites not specified)

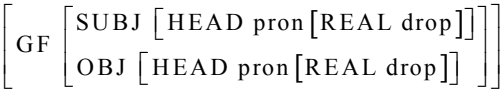
suObOb2Clit = subject and object and object2 are cliticized (cliticization sites not specified)

suOb2Clit = subject and object2 are cliticized (cliticization sites not specified)

suDrop = subject is dropped



suObDrop = subject and object are dropped



suObIobDrop = subject and object and indirect object are dropped

suObOb2Drop = subject and object and object2 are dropped

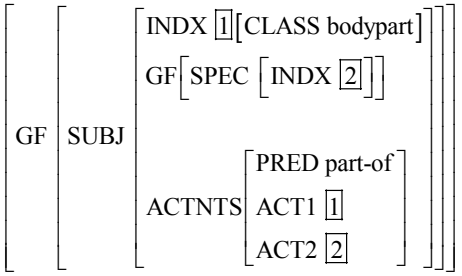
suIobDrop = subject and indirect object are dropped

suOb2Drop = subject and object2 are dropped

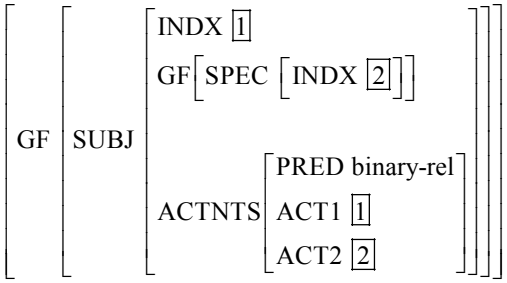
suSpecBodypart = suBPspec

suSpecBP = suBPspec

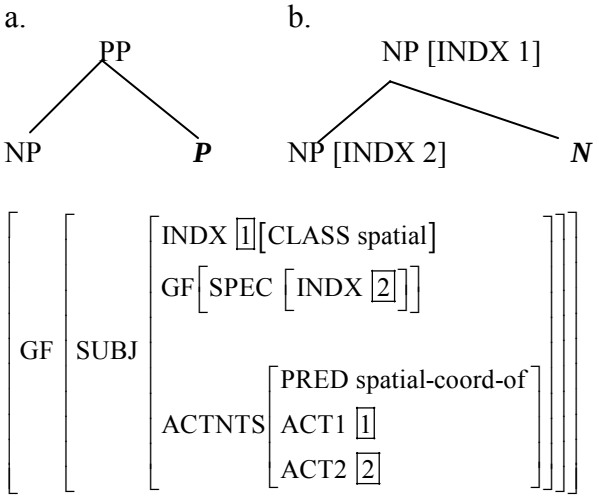
suBPspec = (the referent of) the subject's specifier (literal ex: "his heart", "his head").



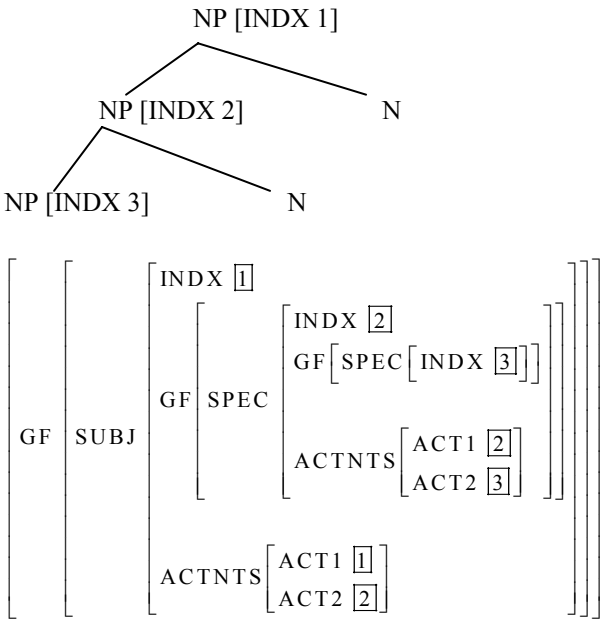
suPossp = the subject has a possessor (NP) phrase as specifier.



suPostp = the subject is a 'postpositional phrase'; that is to say, the subject could be analyzed either as (a) a PP with preposition last (giving the name of the label), or, (b) as reflected in the feature structure below, as an NP with a relational noun as head and an NP specifier (literal ex: "his inside").

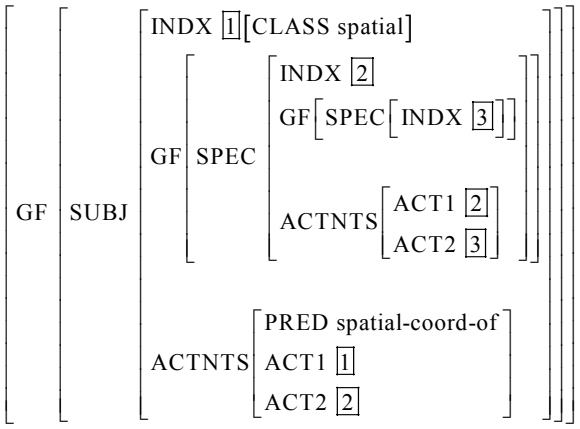


suSpecPossp = the subject's specifier has a possessor NP phrase as specifier (literal ex: "his head's edge")



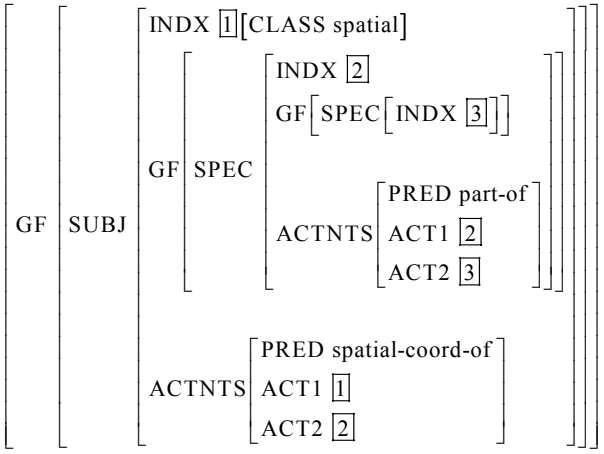
suSpecPostp = the subject's specifier is a postpositional phrase (same tree structure as above)

suPostpSpecPossp (a specialization of **suSpecPossp**) = the subject is a postpositional phrase and has a possessor NP as specifier of its specifier (literal ex: "his head's edge")



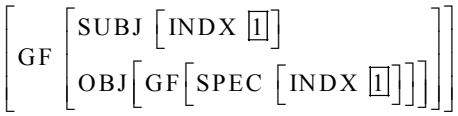
suPostpSpecBodypart = suPostpSpecBP

suPostpSpecBP (a special case of **suPostpSpecPossp**) = the subject is a postpositional phrase and the specifier of its specifier is in a body-bodypart relation to its head (literal ex: "his head's edge")

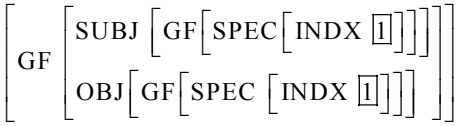


suSpecBPspec = the subject's specifier is a bodypart of the subject's specifier's specifier ("his head's edge") (same logic as with **suBPspec**)

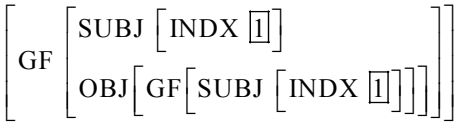
suIDobSpec = the subject is identical to the specifier of the object



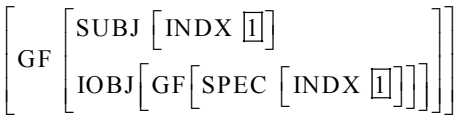
suSpecIDobSpec = the specifier of the subject is identical to the specifier of the object



suIDobSu = the subject is identical to the subject of the object (the object being a clause)



suIDiobSpec = the subject is identical to the specifier of the indirect object



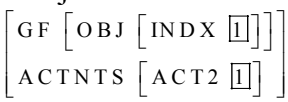
suIDobSpecSpec = the subject is identical to the specifier of the specifier of the object

suIDcompSu = the subject is identical to the complement's subject

For the feature structure of many of the **ob**... labels, see corresponding labels starting with **su.**..

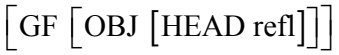
obDir = object is understood in a directional capacity.

obArg = object functions as argument relative to the matrix verb.

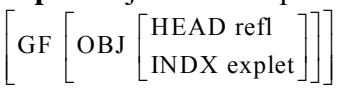


obPro = object is a pronoun.

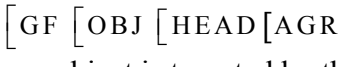
obRefl = object is a reflexive pronoun.



obReflExpl = object is an expletive reflexive pronoun.

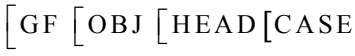


obDECL = object is a declarative clause
obDECLcmp = object is a declarative clause with a complementizer
obDECLbare = object is a declarative clause without a complementizer
obIRR = object is an irrealis clause
obIRRcmp = object is an irrealis clause with a complementizer
obIRRbare = object is an irrealis clause without a complementizer
obYN = object is a yes-no-interrogative clause.
obWH = object is a wh-interrogative clause.
obOM = object is targeted by the verb's object marking



obAgrsc = object is targeted by the secondary predicate's agreement marking

obAcc = object is marked Accusative



obGen = object is marked Genitive
obDat = object is marked Dative
obNom = object is marked Nominative

obAbsl = object has case Absolutive

[GF [OBJ [HEAD [CASE absol]]]]

obDef = object is definite

[GF [OBJ [HEAD [DEF +]]]]

obIndef = object is indefinite

[GF [OBJ [HEAD [DEF -]]]]

obAccDef = object is marked Accusative and is definite

[GF [OBJ [HEAD [CASE acc, DEF +]]]]

obAccIndef = object is marked Accusative and is indefinite

obAccDefOM = object is marked Accusative, is definite, and is targeted by the verb's object marking

[GF [OBJ [HEAD [CASE acc, DEF +, AGR-TARGET +]]]]

obClit = object is cliticized (cliticization site not specified)

[GF [OBJ [HEAD pron [REAL clit]]]]

obIobClit = object and indirect object are cliticized (cliticization sites not specified)

obOb2Clit = object and object2 are cliticized (cliticization sites not specified)

obDrop = object is dropped

[GF [OBJ [HEAD pron [REAL drop]]]]

obIobDrop = object and indirect object are dropped

obOb2Drop = object and object2 are dropped

obPossp = the object has a possessor (NP) phrase as specifier. (See definition of **suPossp**.)

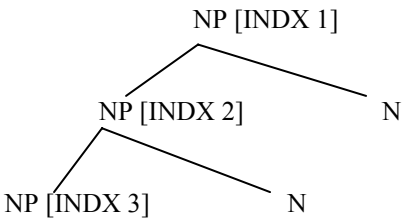
obPostp = the object is a 'postpositional phrase'; that is to say, the object could be analyzed either as (a) a PP with preposition last (giving the name of the label), or, (b) as an NP with a relational noun as head and an NP specifier (literal ex: "his inside"). (See definition of **suPostp**.)

obSpecBodypart = **obBPspec**

obSpecBP = **obBPspec**

obBPspec = (the referent of) the object is a bodypart of (the referent of) the specifier (literal ex: "his heart", "his head"). (See definition of **suBPspec**.)

obSpecPossp = the object's specifier has a possessor NP phrase as specifier (literal ex: "his head's edge")



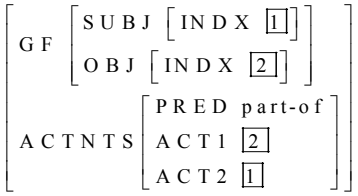
obSpecPostp = the object's specifier is a postpositional phrase (same tree structure as above)

obPostpSpecPossp (a specialization of **obSpecPossp**) = the object is a postpositional phrase and has a possessor NP as specifier of its specifier (literal ex: "his head's edge")

obPostpSpecBodypart = obPostpSpecBP

obPostpSpecBP (a specialization of **obPostpSpecPossp**) = the object is a postpositional phrase and the specifier of its specifier is in a body-bodypart relation to its head (literal ex: "his head's edge")

obPRTOFsu = the referent of the object is interpreted as part-of the referent of the subject.



(Ex. Ga: *E-yɛ tsui*
3S-have heart
'He is patient')

obUnif = object is an 'inherent complement', i.e., unifies with the verb to determine the verbal meaning

obNomvL = object is a nominalization of a verbal expression, in which the verb occurs last ie. following its arguments

obSpecNomvL = object's specifier is a nominalization of a verbal expression, in which the verb occurs last

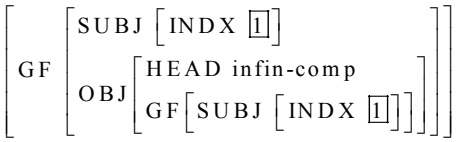
obIDexpnSu = object is identical to extraposed clause's subject

obIDsuSpec = object is identical to the specifier of the subject

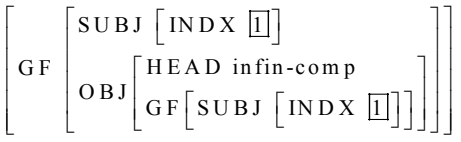
obSpecIDvidObSpec = object's specifier is identical to Verbid's object's specifier

obIDvidObSpec = object is identical to Verbid's object's specifier

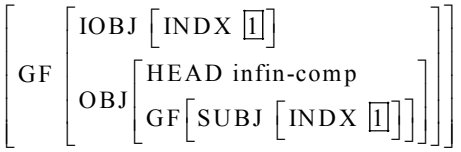
obEqInf = object is an infinitive equi-controlled by the subject (used when there is only one option)



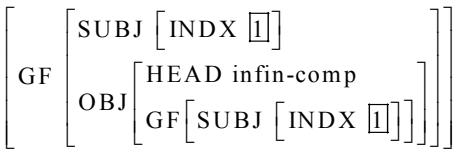
obEqSuInf = object is an infinitive equi-controlled by subject (used when there is more than one option).



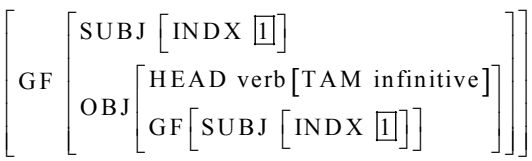
obEqIobInf = object is an infinitive equi-controlled by indirect object (used when there is more than one option).



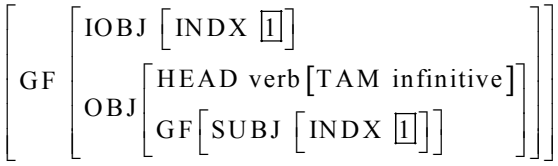
obEqBareinf = object is a bare infinitive equi-controlled by the subject (used when there is only one option)



obEqSuBareinf = object is a bare infinitive equi-controlled by subject (used when there is more than one option).



obEqIobBareinf = object is a bare infinitive equi-controlled by indirect object (used when there is more than one option).



.

obAbsInf = object is a non-controlled ('absolute') infinitive.

obAspIDvAsp = (a clausal object:) object's Aspect is identical to the matrix verb's aspect

For the feature structure of many of the **iob**... labels, see corresponding labels starting with **su.**.. or **ob**…

iobReflExpl = indirect object is an expletive reflexive.
iobOM = indirect object is targeted by the verb's object marking
iobAcc = indirect object is marked Accusative
iobGen = indirect object is marked Genitive
iobDat = indirect object is marked Dative
iobPostp = the indirect object is a postpositional phrase (literal ex: "his inside").
iobCl = indirect object is cliticized (cliticization site not specified)
iobDrop = indirect object is dropped

For the feature structure of many of the **ob2**... labels, see corresponding labels starting with **su.**.. or **ob**…

ob2DECLcmp = second object is a declarative clause with complementizer
ob2OM = second object is targeted by the verb's object marking
ob2Acc = second object is marked Accusative
ob2Gen = second object is marked Genitive
ob2Dat = second object is marked Dative
ob2Unif = object2 is an 'inherent complement', i.e., unifies with the verb to determine the verbal meaning
ob2Cl = object2 is cliticized (cliticization site not specified)
ob2Drop = object2 is dropped

ob2AccDef = object2 is marked Accusative and is definite

ob2AccIndef = object2 is marked Accusative and is indefinite

ob2AccDefOM = object2 is marked Accusative, is definite, and is targeted by the verb's object marking

ob2DatDef = object2 is marked Dative and is definite

ob2DatIndef = object2 is marked Dative and is indefinite

ob2DatDefOM = object2 is marked Dative, is definite, and is targeted by the verb's object marking

Object3 arises in verb extension constructions, typically in languages having little case, so tentatively only the specifications below are relevant.

ob3OM = object3 is targeted by the verb's object marking

ob3Cl = object3 is cliticized (cliticization site not specified)

ob3Drop = object3 is dropped.

Object4 arises exceptionally in verb extension constructions, typically in languages having little case, so tentatively only the specifications below are relevant.

ob4OM = object4 is targeted by the verb's object marking

ob4Cl = object4 is cliticized (cliticization site not specified)

ob4Drop = object4 is dropped

oblRefl = the governee of the oblique is a reflexive.

oblOM = oblique is targeted by the verb's object marking

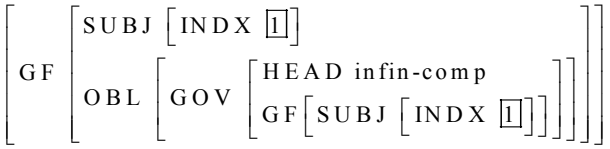
oblDECL = the governee of the oblique is a declarative clause.

oblYN = the governee of the oblique is a yes-no-interrogative clause.

oblWH = the governee of the oblique is a wh-interrogative clause.

oblAbsinf = the governee of the oblique is a non-controlled infinitive.

oblEqSuInf = the governee of the oblique is an infinitive equi-controlled by subject.

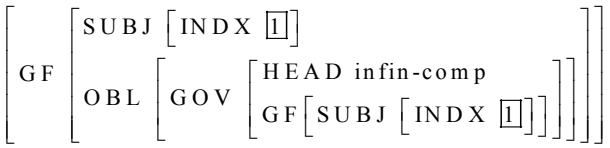


(Ex. No: *han håper på å komme* 'he hopes [on] to come')

oblEqObInf = the governee of the oblique is an infinitive equi-controlled by object.

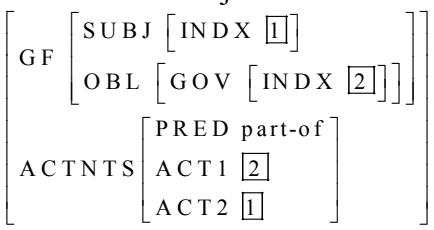
(Ex. Norw: *han bønnfalt meg om å gå*
'he begged me about to go' = “he begged me that I leave”)

oblRaisInf = the governee of the oblique is an infinitive which is raising-controlled by the subject.



(Ex. : Norw. *han later til å komme*
'he appears [to] to come')

oblPRTOFsu = the referent of the governee of the oblique is interpreted as part-of the referent of the subject.



(Ex. : Norw. *han fryser på ryggen*
'he freezes on the back'
= 'his back is cold"

oblPRTOFob = the referent of the governee of the oblique is interpreted as part-of the referent of the object.

oblPRTOFiob = the referent of the governee of the oblique is interpreted as part-of the referent of the indirect object.

oblExlnkAbsinf = extralinked is a non-controlled infinitive occurring as governee of an oblique.

[GF [OBL [GOV [HEAD infin-comp]]]]

[An *extralinked* clause is like an *extraposed* clause except that substituting it for the expletive does not yield a grammatical construction.]

(Ex.:Norw. *det haster med å rydde*
'it hastes with to tidy' = "it is urgent that it gets tidied up")

oblExlnkDECL = extralinked is a declarative clause occurring as governee of an oblique.

[GF [OBL [GOV [HEAD decl-comp]]]]

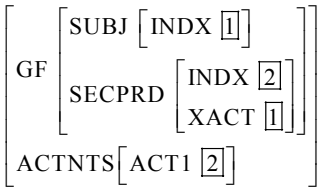
(Ex.: Norw. *det ser ut til at han kommer*
'it looks out to that he comes' = "it seems that he comes")

presDir = presented (NP in presentational) is understood in a directional capacity.

[GF [PRES [INDX [ROLE oriented-obj]]]]

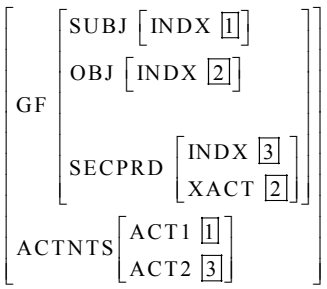
(Ex.: Norw. *det løper en mann*
'there runs a man' = "there is a man running")

scSuNrg = the secondary predicate is predicated of a non-argument subject (i.e., a subject not serving as semantic argument of the matrix verb – a construction sometimes referred to as 'raising to subject').



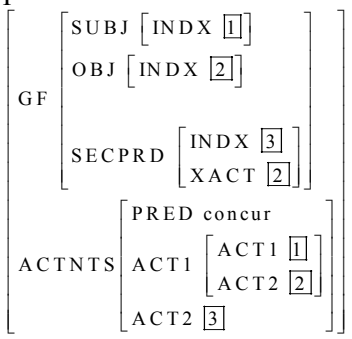
(Ex.: Eng. *he seems sick*)

scObNrg = the secondary predicate is predicated of a non-argument object (i.e., an object not serving as semantic argument of the matrix verb – a construction sometimes referred to as 'raising to object').



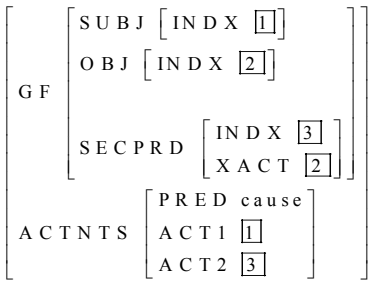
(Ex.: Eng. *I saw him sleeping*)

scObArgConcur = the secondary predicate is predicated of an argument object (i.e., an object serving as semantic argument of the matrix verb), and the matrix verb (together with its subject) is part of the description of an event concurrent with the situation described by the secondary predication.



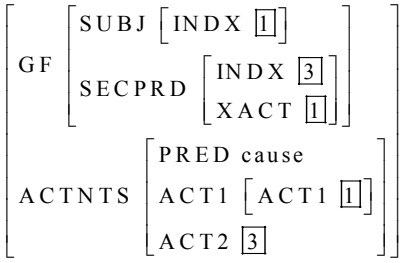
(Ex.: Eng. *he drank the coffee warm*)

scObNrgRes = with a person-causer, a one-actant caused event (incrementally or not), and the XACT of the predicative expressed as object (the object is not serving as semantic argument of the matrix verb). (Compare with **trCs**, section 5.)



(Ex.: Eng. *he made the horse jump*)

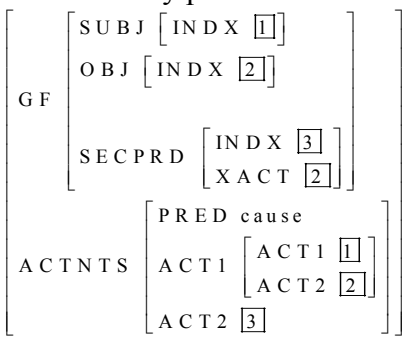
scSuArgCsd = the secondary predicate is predicated of an argument subject (i.e., a subject serving as semantic argument of the matrix verb), and the matrix verb (together with its subject) is part of the description of an event causing the situation described by the secondary predication.



(Ex.: Norw. *kaffen koker bort*
'the coffee boils away')

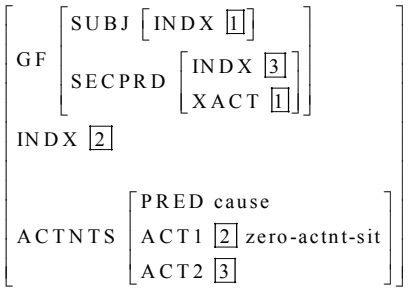
scResIncrm = **scSuArgCsd** (with causation understood as being incremental)
scSuNrgResIncrm = **scSuNrgCsd** (with causation understood as being incremental)

scObArgCsd = the secondary predicate is predicated of an argument object (i.e., an object serving as semantic argument of the matrix verb), and the matrix verb (together with its subject) is part of the description of an event causing (incrementally or in one event) the situation described by the secondary predication.



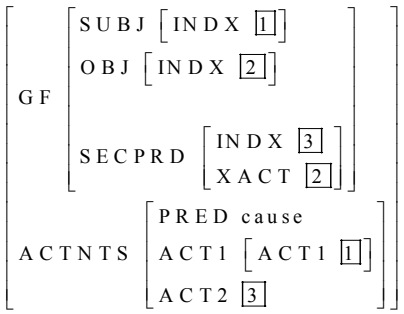
(Ex.: Eng. *he kicked the ball flat*)

scSuNrgCsd = the secondary predicate is predicated of a non-argument subject (i.e., a subject not serving as semantic argument of the matrix verb – "raising to subject'), and the matrix verb is part of the description of an event causing the situation described by the secondary predication.



(Ex.: Norw. *landsbyen snør ned*
'the village snows down'
ie. 'the village gets snowed in')

scObNrgCsd = the secondary predicate is predicated of a non-argument object (i.e., an object not serving as semantic argument of the matrix verb – "raising to object'), and the matrix verb (together with its subject) is part of the description of an event causing the situation described by the secondary predication.



(Ex.: Norw. *han sang rommet tomt*
'he sang the room empty')

For all the causative labels, there is a possible final specification:

..**rgCsdZero** = ... the matrix verb is part of the description of a zero-participant event causing the situation described by the secondary predication.

..**rgCsdUnar** = ... the matrix verb is part of the description of a one-participant event causing the situation described by the secondary predication.

..**rgCsdBinar** = ... the matrix verb is part of the description of a two-participant event causing the situation described by the secondary predication.

scAdj = the secondary predicate is headed by an adjective
[GF [SECPRD [HEAD adj]]]

scAdj_agr = the secondary predicate is headed by an adjective which carries an agreement formative
[GF [SECPRD [HEAD adj[FORMATIVES ⟨AGR⟩]]]]

scN = the secondary predicate is headed by a noun

scPP = the secondary predicate is a PP

scPrtcl = the secondary predicate is a particle

scAdv = the secondary predicate is headed by an adverb
scPredprtcl = the secondary predicate is headed by a predparticle
scInf = the secondary predicate is an infinitive clause
scBareinf = the secondary predicate is a bare infinitive clause
scPerf = the secondary predicate is a perfective phrase
scEquat = the secondary predicate is an equative phrase

compDECL = complement is a declarative clause.
[GF [COMP [HEAD decl-comp]]]
compDECLbare = complement is a declarative clause without complementizer
compYN = complement is a yes-no-interrogative clause.
compWH = complement is a wh-interrogative clause.
compIRR = complement is an irrealis clause
compIRRcmp = complement is an irrealis clause with a complementizer

expnDECL = a declarative clause is extraposed.
[GF [EXPN [HEAD decl-comp]]]
expnYN = a yes-no-interrogative clause is extraposed.
expnWH = a wh-interrogative clause is extraposed.
expnCOND = a conditional clause is extraposed.
expnEqInf = an equi-controlled infinitive is extraposed.
expnAbsinf = a non-controlled infinitive is extraposed.
expnInfabs = a non-controlled infinitive is extraposed.
expnHYP = a hypothetical clause is extraposed.
expnEQUAT = an equative clause is extraposed.
exlnkDECL = a declarative clause is extralinked.
[GF [GOV [EXLNK [HEAD decl-comp]]]]
exlnkYN = a yes-no-interrogative clause is extralinked.
exlnkWH = a wh-interrogative clause is extralinked.
exlnkCOND = a conditional clause is extralinked.
exlnkEqInf = an equi-controlled infinitive is extralinked.
exlnkAbsinf = a non-controlled infinitive is extralinked.

Derivational (operational) specifications

These specifications trace the derivational history of a GF, in a way similar to 'chains' in GB and Relational Grammar.
For discussion and illustration, see **section V**.

(For effects of morphological causativization:)
`obCsu` = **ob** which would have been *su* relative to *input* of *Causative* formation
`obCob` = **ob** which would have been *ob* relative to *input* of *Causative* formation
`obCob2` = **ob** which would have been *ob2* relative to *input* of *Causative* formation
`obCiob` = **ob** which would have been *iob* relative to *input* of *Causative* formation
`obCobl` = **ob** which would have been *obl* relative to *input* of *Causative* formation

`ob2Csu` = **ob2** which would have been *su* relative to *input* of *Causative* formation

ob2Cob = **ob2** which would have been *ob* relative to *input* of *Causative* formation
ob2Cob2 = **ob2** which would have been *ob2* relative to *input* of *Causative* formation
ob2Cobl = **ob2** which would have been *obl* relative to *input* of *Causative* formation

iobCsu = **iob** which would have been *su* relative to *input* of *Causative* formation
iobCob = **iob** which would have been *ob* relative to *input* of *Causative* formation
iobCiob = **iob** which would have been *iob* relative to *input* of *Causative* formation
iobCobl = **iob** which would have been *obl* relative to *input* of *Causative* formation

oblCsu = **obl** which would have been *su* relative to *input* of *Causative* formation
oblCob = **obl** which would have been *ob* relative to *input* of *Causative* formation
oblCob2 = **obl** which would have been *ob2* relative to *input* of *Causative* formation
oblCiob = **obl** which would have been *iob* relative to *input* of *Causative* formation
oblCobl = **obl** which would have been *obl* relative to *input* of *Causative* formation

(For the promotional part of Passive formation:)
suPob = **su** which would have been *ob* relative to *input* of *Passive* formation
suPob2 = **su** which would have been *ob2* relative to *input* of *Passive* formation
suPiob = **su** which would have been *iob* relative to *input* of *Passive* formation
suPobl = **su**which would have been *obl* relative to *input* of *Passive* formation

(For the promotional part of *Stative* formation:)
suSob = **su** which would have been *ob* relative to *input* of *Stative* formation

(For the promotional part of *Middle* formation:)
suMob = **su** which would have been *ob* relative to *input* of *Middle* formation

(For the promotional part of *Applicative* formation:)
obAobl = **ob** which would have been *obl* relative to *input* of *Applicative* formation
iobAobl = i**ob** which would have been *obl* relative to *input* of *Applicative* formation
ob2Aobl = **ob2** which would have been *obl* relative to *input* of *Applicative* formation

(Repercussion effects:)
obUob2 = *ob* 'up from' *ob2* (because old *ob* has disappeared (promoted, deleted,...))
ob2Uob3 = *ob2* 'up from' *ob3* (because old *ob2* has disappeared)
ob3Uob4 = *ob3* 'up from' *ob4* (because old *ob3* has disappeared)
ob2Dob = *ob2* 'down from' *ob* (because a new *ob* has appeared)
ob3Dob2 = *ob3* 'down from' *ob2* (because a new *ob2* has appeared)
ob4Dob3 = *ob4* 'down from' *ob3* (because a new *ob3* has appeared)

('Absorption' effects:)
`nilRob =` ob is 'absorbed' through Reflexivization
`nilRPob =` ob is 'absorbed' through Reciprocization

(Recursion:)
`suPobCsu` = **`su`** which would have been *ob* relative to *input* of *Passive* formation - an **`ob`** which would have been *su* relative to *input* of *Causative* formation
`suPobCob` = **`su`** which would have been *ob* relative to *input* of *Passive* formation - an **`ob`** which would have been *ob* relative to *input* of *Causative* formation
`suPobCob2` = **`su`** which would have been *ob* relative to *input* of *Passive* formation - an **`ob`** which would have been *ob2* relative to *input* of *Causative* formation
`suPobCiob` = **`su`** which would have been *ob* relative to *input* of *Passive* formation - an ob which would have been *iob* relative to *input* of *Causative* formation
`suPobCobl` = **`su`** which would have been *ob* relative to *input* of *Passive* formation – an **`ob`** which would have been *obl* relative to *input* of *Causative* formation

`suPob2Csu` = **`su`** which would have been *ob2* relative to *input* of *Passive* formation - an **`ob2`** which would have been *su* relative to *input* of *Causative* formation
`suPob2Cob` = **`su`** which would have been *ob2* relative to *input* of *Passive* formation - an **`ob2`** which would have been *ob* relative to *input* of *Causative* formation
`suPob2Cob2` = **`su`** which would have been *ob2* relative to *input* of *Passive* formation - an **`ob2`** which would have been *ob2* relative to *input* of *Causative* formation
`suPob2Ciob` = **`su`** which would have been *ob2* relative to *input* of *Passive* formation - an ob2 which would have been *iob* relative to *input* of *Causative* formation
`suPob2Cobl` = **`su`** which would have been *ob2* relative to *input* of *Passive* formation – an **`ob2`** which would have been *obl* relative to *input* of *Causative* formation

`suPiobCsu` = **`su`** which would have been i*ob* relative to *input* of *Passive* formation - an **`iob`** which would have been *su* relative to *input* of *Causative* formation
`suPiobCob` = **`su`** which would have been i*ob* relative to *input* of *Passive* formation - an **`iob`** which would have been *ob* relative to *input* of *Causative* formation
`suPiobCob2` = **`su`** which would have been i*ob* relative to *input* of *Passive* formation - an **`iob`** which would have been *ob2* relative to *input* of *Causative* formation

suPiobCiob = **su** which would have been i*ob* relative to *input* of *Passive* formation - an iob which would have been *iob* relative to *input* of *Causative* formation

suPiobCobl = **su** which would have been i*ob* relative to *input* of *Passive* formation – an i**ob** which would have been *obl* relative to *input* of *Causative* formation

suPoblCsu = **su** which would have been *obl* relative to *input* of *Passive* formation - an **obl** which would have been *su* relative to *input* of *Causative* formation

suPoblCob = **su** which would have been *obl* relative to *input* of *Passive* formation - an **obl** which would have been *ob* relative to *input* of *Causative* formation

suPoblCob2 = **su** which would have been *obl* relative to *input* of *Passive* formation - an **obl** which would have been *ob2* relative to *input* of *Causative* formation

suPoblCiob = **su** which would have been *obl* relative to *input* of *Passive* formation - an obl which would have been *iob* relative to *input* of *Causative* formation

suPoblCobl = **su** which would have been *obl* relative to *input* of *Passive* formation – an **obl** which would have been *obl* relative to *input* of *Causative* formation

suPobAobl = **su** which would have been *ob* relative to *input* of *Passive* formation – an **ob** which would have been *obl* relative to *input* of *Applicative* formation

suPob2Aobl = **su** which would have been *ob2* relative to *input* of *Passive* formation – an **ob2** which would have been *obl* relative to *input* of *Applicative* formation

suPiobAobl = **su** which would have been i*ob* relative to *input* of *Passive* formation – an i**ob** which would have been *obl* relative to *input* of *Applicative* formation

suRAISsuMob = subject is raised from subject, and before that promoted thereto from object by Middle Formation

obRAISsuMob = object is raised from subject, and before that promoted thereto from object by Middle Formation

III. Slots 4, 5 and 6 in Single-verb constructions: Role, Aspect/Aktionsart, and Situation type

III.A. SLOT 4 Roles

Except for 'Abst' and 'Sit', which mark a specific ontological type, there are no capped parts of role labels. When used, the role label is prefixed by a grammatical function, so that, e.g., 'ag' occurs as 'suAg'. Another example:

> **vidObEndpt** = the role of the object in the Verbid phrase is 'endpoint'
> [GF [VID [GF [OBJ [INDX[ROLE endpoint]]]]]]

```
activated = item set into some activity
aff = affected
affincrem = incrementally affected
ag = agent
agintent = agent relative to intended/considered eventuality
agmover = agentive mover
agsens = agentive senser
alongline  = line being followed
ass = assessor
ben = beneficiary / benefactive
cog = cognizer
com = commitative
content = content of thought/ communication
csd = caused
csee = causee
csr = causer
dir = directional
distunit = distance unit of movement/extension
ejct = ejected
effector = item effecting
endpt = endpoint of movement/extension
endstate = endstate of development
eventtunit = event unit of activity/ eventuality
exp = experiencer
idfd = item identified in an identity predication
idfng = item providing identification in an identity predication
instr = instrument
interloc = interlocutor
loc = location
locth = locative theme
locus = locus of event
mal = malefactive
mover = locomotor
orientedline = line being oriented
orientedobj = instance of movement/extension
path = path/trajectory of movement/extension
prcpt = percept
permissee = one given permission to do something
poss = possessor
possAbst = abstract possessor
possd = possessed
pres = presented
quality = ascribed quality
rec = recipient
sens = senser
```

startpt = startpoint of movement/extension
trgt = target of attention
time = timepoint or timespan of activity/ eventuality
timeunit = measured time unit of activity/ eventuality
th = theme
thAbst = abstract theme
thincrem = theme incrementally involved
thmover = theme mover
thSit = situational theme
thvehcl = vehicle
top = topic
viapt = viapoint of movement/extension
weightunit = unit of ascribed weight
xBPy = x is a body part of y

SLOT 5 Aspect/ Aktionsart

ACHVMNT
ACT
ACTIVATION
ACTIVITY
COMPLETED_MONODEVMNT
COMPLETED_ACTIVITY
COMPLETION
EVENT
GEN
HAB
INCH
INCHOATION
INCREMRESULT
ITER
NONCOMPLETED
NONCOMPLETED_INST
NONCOMPLETED_MONODEVMNT
NONCOMPLETED_MONODEVMNT_MEDIUM
PHENOM_TELIC
PROCESS
PROTR
SEMELFACTIVE
STATIVE
TELIC

SLOT 6 Situation Type

(With each situation type are entered roles specific to that situation type, to be referred to in slot 7, the slot for ‘translational linking’.)

ABOUTNESS (CONTENT, REFERENCE)
ACCOMPANYING (MOVER, MOVERACCOMPANIED)
ACQUISITION (AGENT, ACQUIRED)
AFFECT (EFFECTOR, AFFECTED)
ALONGLINEEXTENSION (EXTENDEDOBJ, LINEFOLLOWED)
ALONGLINEMOTION (MOVER, LINEFOLLOWED)
ASKING_ABOUT (AGENT, INTERLOCUTOR, QUERYMATTER)
ASSESS (AGENT, ASSESSEDMATTER)
AVAILING (AGENT, BENEFICIARY, UNDERGOER)
CARETAKING
CAUSATION_WITH_CAUSINGENTITY (CAUSER, CAUSED)
CAUSATION_WITH_CAUSINGEVENT (CAUSE, CAUSED)

```
CAUSE_RESULT  (CAUSE, RESULT)
COGN  (COGNIZER, COGNCONTENT)
COGNITION (COGNIZER, COGNCONTENT)
COLLECT  (ACTOR, UNDERGOER, [CONSTRUCTEDENTITY])
COMMITMENT  (AGENT, COMMITMATTER)
COMMUNICATION  (AGENT, CONTENT, INTERLOCUTOR, REFERENCE)
COMPARISON  (AGENT, COMPARANDUM, REFERENCE)
COMPARISON_COMPARATIVE  (AGENT, COMPARANDUM, REFERENCE)
COMPARISON_EQUATIVE  (AGENT, COMPARANDUM, REFERENCE)
CONCURRSTATE  (CONCURRINGSTATE)
CONTINUATION
CONTACTEJECTION  (LAUNCHER, MOVER, TARGET) (Mover keeps contact with
Launcher during the whole act, and attains contact with Target at the end
of the act)
COVER (COVER, AREACOVERED)
CROSSINGMOTIONS  (MOVER, MOVERCROSSED)
CROSSINGPATHS (EXTENDEDOBJ, LINECROSSED)
CUTTING (ACTOR, INSTRUMENT, AFFECTED, [CONSTRUCTEDENTITY])
DEPEND  (DEPENDENT, DEPENDABLE)
DOFREQUENTLY ()
EJECT  (EJECTOR, EJECTED)
EJECTION (EJECTOR, EJECTED)
EJECTION_DIRECTED  (EJECTOR, EJECTED, ORIENTATION)
EMOTION  (EXP, [EXPERIENCED])
EMOTION_CAUSED (CAUSE, EXP)
EMOTION_DIRECTED  (EXP, [EXPERIENCED], ORIENTATION)
EMOTION_TARGETED (EXP, TARGET)
ENDPT_EXTENSION (EXTENDEDOBJ, ENDPOINT) +asp
ENDPT_MOTION  (MOVER, ENDPOINT) +asp
EPISODIC_PROPTY  (ASCR) +asp
EXHIBACT
EXHIBPROPTY
EXPER (EXPERIENCER, EXPERIENCED)
EXPERIENCING_PROTR  (EXPERIENCER, EXPERIENCED) +asp
EXTENDING
FINISH
HELP
IMPRECATION
IDENTITY
INTENT
LASTING
LINESITUATING  (ACTOR, EXTENDEDOBJ, ORIENTATION)
LOCATION  (ITEMLOCATED, LOCATION)
LOCOMOCONDUCTION
LOCUTACT  (LOCUTOR, CONTENT, INTERLOCUTOR, REFERENCE)
MAINTAINPOSITION (MAINTAINER, POSITION) +asp
MAINTAINSTATE (MAINTAINER, STATE)
MALEFACTION
MENTION (MENTIONER, MENTIONED)
MOTION  (MOVER)
MOTION_CAUSED (CAUSE[R], MOVER)
MOTION_DIRECTED (MOVER, ORIENTATION)
NEED
OPINION
ORIENTING (ORIENTEDOBJ) +asp
PARTWHOLE_AFFECTING (EFFECTOR, WHOLEAFFECTED, PARTAFFECTED)
PENDINGSTATE  (ASCR) +asp
PERCPT
PERFORM
PERFORMANCE
PERFORMFUNCTION
```

```
PERMISSION
PHENOM
PLACEMENT  (EFFECTOR, UNDERGOER, ENDPOSITION)
POSSESS  (POSSESSOR, POSSESSED)
POSTURE_LOC  (POSTURED, LOCATION) +asp
PRESENTATION  (PRESENTED)
PROPOSITIONALATTITUDE  (ASCR)
PROPOSITIONALATTITUDE_ACTIVITY (ASCR) +asp
PROPTY (ASCR)
PROPTY_DYN  (ASCR) +asp
PROPTY_DYN_ACQUIRD (ASCR)
PROPTY_DYN_ESTBLSHD (ASCR)
PROPTY_DYN_PROGR (ASCR) +asp
PROPTY_ESTABD (ASCR)
PROPTY_GEN (ASCR) +asp
PROPTY_PROGR (ASCR) +asp
PSYCHSTATE (ASCR) +asp
REDUCTION
REMOVAL  (EFFECTOR, UNDERGOER, DEPLETEDPOSITION)
RENDERING_IN_POSITION (EFFECTOR, UNDERGOER, ENDPOSITION)
REPRESENT  (REPRESENTER, REPRESENTED)
SENS  (SENSER, PERCEIVED)
SENSING_PROTR (SENSERE, PERCEIVED) +asp
STATE  (ASCR)
SUSTAINEDACTIVITY  (ACTOR) +asp
SUSTAINEDSTATE  (ASCR) +asp
TRANSFER  (INSTIGATOR, UNDERGOER, ENDPOSSESSOR) +asp
TRANSFER_PURPOSE
USINGPATH  (ACTOR, PATH)
USINGVEHICLE  (ACTOR, VEHICLE)
VIAPT_EXTENSION  (EXTENDEDOBJ, VIAPOINT) +asp
VIAPT_MOTION  (MOVER, VIAPOINT) +asp
WASHING  (ACTOR, UNDERGOER)
WEIGHING  ([ACTOR], UNDERGOER, MEASURE)
```

IV. Template architecture for Multi-verb constructions

This section addresses 4 types of multiverb constructions:
- Serial Verb Constructions (SVC, label: sv)
- Extended Verb Complexes (EVC, label: ev)
- Auxiliary Verb Constructions (AVC, label: axv)
- Verbids (VID, label: vid)

Some of those instantiate phenomena named 'Complex Predicates' in the literature, however the notions only partially intersect: not all Complex Predicates involve multiple verbs, and not all of the four types listed here would fall under the notion 'Complex Predicate'. Auxiliary Verb Constructions and Extended Verb Complexes have much in common and will be treated under the same heading.

A Serial Verb Constructions.

These are represented with three major areas: first a 'global' code indicating sv status together with the number of verbs in the series, and possible identities holding all across the series; second, information bits about the various verbs' valence and arguments of the verbs; and second, a situation type label covering the whole construction. The first and third specifications are short, whereas the specifications in the second area can constitute a long string. Area 3 is not exemplified here.

AREA 1 Global construction labels

For up to 4-membered series, the global labels are:

sv = serial verb construction with 2 members

$$\begin{bmatrix} \text{V1} & [\text{HEAD verb}] \\ \text{V2} & [\text{HEAD verb}] \end{bmatrix}$$

sv3 = serial verb construction with 3 members
sv4 = serial verb construction with 4 members
sv_suID = serial verb construction with 2 members and shared reference between the subjects of the verbs

$$\begin{bmatrix} \text{V1} & \left[\text{GF}\left[\text{SUBJ}\left[\text{INDX } \boxed{1}\right]\right]\right] \\ \text{V2} & \left[\text{GF}\left[\text{SUBJ}\left[\text{INDX } \boxed{1}\right]\right]\right] \end{bmatrix}$$

sv3_suID = serial verb construction with 3 members and shared reference between the subjects
sv4_suID = serial verb construction with 4 members and shared reference between the subjects
sv_obID = serial verb construction with 2 members and shared reference between the objects

$$\begin{bmatrix} \text{V1} & \left[\text{GF}\left[\text{OBJ}\left[\text{INDX } \boxed{1}\right]\right]\right] \\ \text{V2} & \left[\text{GF}\left[\text{OBJ}\left[\text{INDX } \boxed{1}\right]\right]\right] \end{bmatrix}$$

sv3_obID = serial verb construction with 3 members and shared reference between the objects
sv4_obID = serial verb construction with 4 members and shared reference between the objects

sv_aspID = serial verb construction with 2 members and shared aspectual value

$$\begin{bmatrix} V1 \begin{bmatrix} ASPECT \boxed{1} \end{bmatrix} \\ V2 \begin{bmatrix} ASPECT \boxed{1} \end{bmatrix} \end{bmatrix}$$

sv3_aspID = serial verb construction with 3 members and shared aspectual value
sv4_aspID = serial verb construction with 4 members and shared aspectual value

sv_suObID = serial verb construction with 2 members and shared reference between the subjects and objects

$$\begin{bmatrix} V1 \begin{bmatrix} GF \begin{bmatrix} SUBJ \begin{bmatrix} INDX \boxed{1} \end{bmatrix} \\ OBJ \begin{bmatrix} INDX \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ V2 \begin{bmatrix} GF \begin{bmatrix} SUBJ \begin{bmatrix} INDX \boxed{1} \end{bmatrix} \\ OBJ \begin{bmatrix} INDX \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

sv3_suObID = serial verb construction with 3 members and shared reference between the subjects and objects
sv4_suObID = serial verb construction with 4 members and shared reference between the subjects and objects
sv_suAspID = serial verb construction with 2 members and shared reference between the subjects and shared aspectual value
sv3_suAspID = serial verb construction with 3 members and shared reference between the subjects and shared aspectual value
sv4_suAspID =. serial verb construction with 4 members and shared reference between the subjects and shared aspectual value
sv_suObAspID = serial verb construction with 2 members and shared reference between the subjects and objects and shared aspectual value

$$\begin{bmatrix} V1 \begin{bmatrix} GF \begin{bmatrix} SUBJ \begin{bmatrix} INDX \boxed{1} \end{bmatrix} \\ OBJ \begin{bmatrix} INDX \boxed{2} \end{bmatrix} \end{bmatrix} \\ ASPECT \boxed{3} \end{bmatrix} \\ V2 \begin{bmatrix} GF \begin{bmatrix} SUBJ \begin{bmatrix} INDX \boxed{1} \end{bmatrix} \\ OBJ \begin{bmatrix} INDX \boxed{2} \end{bmatrix} \end{bmatrix} \\ ASPECT \boxed{3} \end{bmatrix} \end{bmatrix}$$

sv3_suObAspID = serial verb construction with 3 members and shared reference between the subjects and objects and shared aspectual value
sv4_suObAspID = serial verb construction with 4 members and shared reference between the subjects and objects and shared aspectual value

Continuing specifications (entailing that su and asp are shared throughout the series):
_suAg = the subjects in whole series are agentive
_aspPerf = aspect throughout the whole series is Perfective
(e.g., **sv3_suObAspID_suAg_obTh_aspPerf**)

AREA 2. **Specifications relative to each constituent verb construction**

Valence specifications for each verb construction in the series:

v1intr = verb construction 1 is intransitive
[V1 intr]
v2intr = verb construction 2 is intransitive
v3intr = verb construction 3 is intransitive
v4intr = verb construction 4 is intransitive
v1tr = verb construction 1 is transitive
v2tr = verb construction 2 is transitive
v3tr = verb construction 3 is transitive
v4tr = verb construction 4 is transitive
v1ditr = verb construction 1 is ditransitive
v2ditr = verb construction 2 is ditransitive
v3ditr = verb construction 3 is ditransitive
v4ditr = verb construction 4 is ditransitive

Specification relative to arguments inside each verb construction
The general pattern is using the full range of **Slot 3 & 4** labels prefixed by '**v1**', '**v2**' etc; ex.:
v1suAg = the subject of verb construction 1 (V1) is an Agent
[V1 [GF[SUBJ[INDX[ROLE agent]]]]]
(and likewise for all Vx and all GFs and roles)

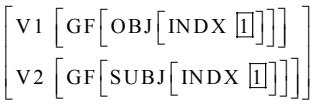
v1aspPerf = the aspect of V1 is Perfective

E.g.:
v1obTh, v1iobBen, v1obEndpt, v1obMover, v2suAg, v2obTh, v2iobBen, v2obEndpt, v3suAg, v3obTh, v3iobBen, v4suAg, v4obTh, v4iobBen, v1aspAor, v1aspPerf, v1suIDobSpec

Of particular relevance:
v2suSM = the subject of V2 is targeted by subject agreement on the verb
v2suClit = the subject of V2 is realized as a cliticized pronoun
[V1 [GF[SUBJ[HEAD[REAL clit]]]]]
(and likewise for v3 and v4:
v3suSM, v3suClit, v4suSM, v4suClit

Identities across specific Vs:
v1suIDv2su = the subject of V1 shares referent with subject of V2
[V1 [GF[SUBJ[INDX 1]]]
V2 [GF[SUBJ[INDX 1]]]]

v1obIDv2su = the object of V1 shares referent with subject of V2 ("switch sharing")



v2aspIDv3asp = the aspect of V2 is identical to aspect of V3
v2suIDv3su = the subject of V2 is identical to subject of V3
v3suIDv4su = the subject of V3 is identical to subjpect of V4
v2obIDv3su = the object of V2 is identical to subject of V3
v3obIDv3su = the object of V3 is identical to subject of V4
v1aspIDv2asp = the aspect of V1 is identical to aspect of V2
v3aspIDv4asp = the aspect of V3 is identical to aspect of V4

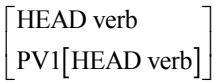
B. Pre-verbal complexes

B.1 Extended Verb Complexes (EVCs; ev)

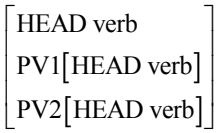
Extended verb complexes act as single verbs relative to the environment, but consist of a limited number of **preverbs** (**pv**) together with the **main verb**. The valence of the main verb determines the valence of the ev relative to the containing clause. Most preverbs are intransitive, but some can be transitive. Conventions for enumerating the preverbs of an ev can be similar to those for enumerating verbs of an sv, although since the range of combinations in an ev is very limited, a small number of labels covering the totality of combinations is more correct. Since these labels will be language dependent, for convenience we here still use the numbered labels, with the proviso that, e.g. in Ga, 'ev2' can stand for two fixed combinations: *kɛ*+deictic, and neg+deictic, and a more accurate label than '2' will be chosen in the actual inventory for Ga.

AREA 1 Global labels

ev = ev with one preverb and the main verb



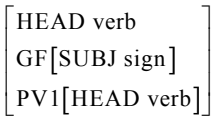
ev2 = ev with two preverbs and the main verb



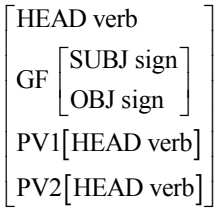
ev3 = ev with three preverbs and the main verb

Defining for the main verb inside an ev what valence it has, and thus the valence of the whole ev as such relative to its environment:

ev_intr = ev with one preverb and an intransitive main verb

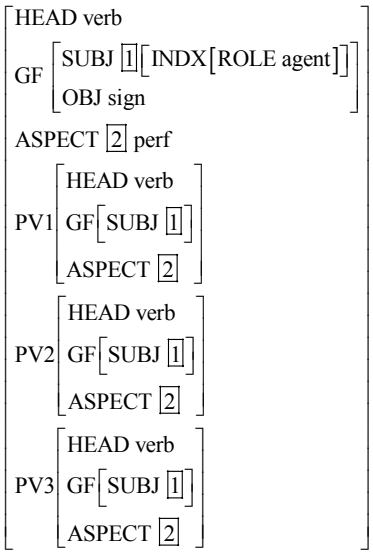


ev2_intr = ev with two preverbs and an intransitive main verb
ev3_intr = ev with three preverbs and an intransitive main verb
ev_tr = ev with one preverb and a transitive main verb
ev2_tr = ev with two preverbs and a transitive main verb



ev3_tr = ev with three preverbs and a transitive main verb
ev_ditr = ev with one preverb and a ditransitive main verb
ev2_ditr = ev with two preverbs and a ditransitive main verb
ev3_ditr = ev with three preverbs and a ditransitive main verb

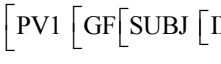
Identities spanning the whole ev are expressed as for svs, e.g.:
ev3_tr_suAspID_suAg_aspPerf = ev with three preverbs and a transitive main verb, where all verbs share subject reference and aspect, and where the role of the subject relative to all the verbs is Agent and the aspect of all the verbs is Perfective



AREA 2. **Specifications relative to each constituent preverb**
Valence specifications for each verb construction in the series:
pv1intr = preverb 1 is intransitive

pv2intr = preverb 2 is intransitive
pv3intr = preverb 3 is intransitive
pv1tr = preverb 1 is transitive
pv2tr = preverb 2 is transitive
pv3tr = preverb 2 is transitive
For the **main** verb, specifications are as in **Slot 1** for ordinary constructions

Specification relative to arguments inside each verb construction
The general pattern is using the full range of **Slot 3** labels prefixed by '**pv1**', '**pv2**' etc; ex.:
pv1suAg = the subject of preverb 1 (PV1) is an Agent



(and likewise for all PVx and all GFs and roles)
pv1aspPerf = the aspect of PV1 is Perfective

Of particular relevance:
pv2suSM = the subject of PV2 is targeted by subject agreement on the verb
pv2suClit = the subject of PV2 is realized as a cliticized pronoun
(and likewise for pv1, and pv3)

Identities across specific PVs:
pv1suIDpv2su = the subject of PV1 shares referent with subject of PV2

$$\begin{bmatrix} \text{PV1} \left[\text{GF}\left[\text{SUBJ}\left[\text{INDX } \boxed{1}\right]\right]\right] \\ \text{PV2} \left[\text{GF}\left[\text{SUBJ}\left[\text{INDX } \boxed{1}\right]\right]\right] \end{bmatrix}$$

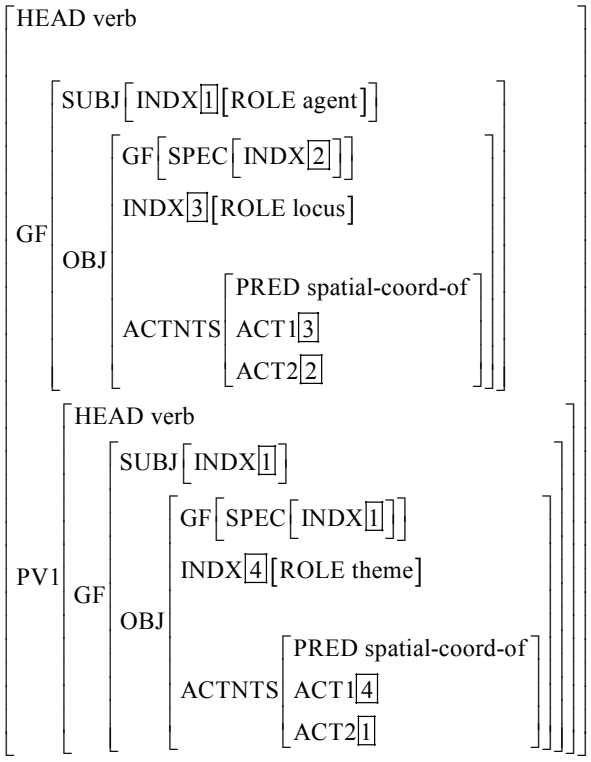
pv2aspIDpv3asp = the aspect of PV2 is identical to aspect of PV3

Frequently used specifications for PVs:
pv1obPro
pv1obInstr
pv1obNomvL
pv1obPossp
pv1suIDpv1obSpec
pv1obThsit
pv1obInstr

Example from Ga:
ev_tr_suAg-pv1tr-pv1obPossp_pv1suIDpv1obSpec-pv1obTh-vtr-obPostp-obLocus

E-kɛ e-hiɛ fɔ-ɔ o-nɔ
3S-move 3S-face throw-HAB 2S.POSS-surface
'She trusts you.'

Explanation:
Ev with one preverb and a transitive main verb and a shared Agent role for subjects of both verbs; PV1 (the only pv) is transitive and its object is a **possp** (see **slot 3**); relative to PV1, its subject is identical to the specifier of the postp object; the object of PV1 has a Theme role and stands in a part-whole relation to its specifier ('her face' being part of 'her'); the main verb is transitive (redundantly specified) and its object - which is the object of the whole verbal complex, and therefore having no prefix on 'ob' - is a postp, and semantically in a part-whole relation to its specifier ('your surface' being a part of 'you'); moreover the object has a Locus role relative to the main verb (the implicit item thrown - the face - ending on 'your surface').
Its AVM:



EVs in SVs

When an ev occurs as a verbal constituent of an sv, the general pattern of sv specification is followed, but the ev status is marked as follows:

Instead of the specifications at the beginning of AREA2 as seen earlier:

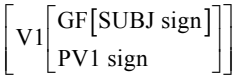
v1intr = verb construction 1 is intransitive

v2intr = verb construction 2 is intransitive

v3intr = verb construction 3 is intransitive ...

one writes:

v1_ev_intr = verb construction 1 is intransitive and the verbal head is constituted by an ev with one preverb



v1_ev2_intr = verb construction 1 is intransitive and the verbal head is constituted by an ev with two preverbs

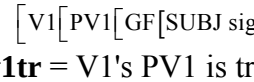
v1_ev3_intr = verb construction 1 is intransitive and the verbal head is constituted by an ev with three preverbs

v2_ev_intr = verb construction 2 is intransitive and the verbal head is constituted by an ev ....

v3_ev_intr = verb construction 3 is intransitive and the verbal head is constituted by an ev ....

For specification of each preverb in an sv, one writes

v1_pv1intr = V1's PV1 is intransitive



v1_pv1tr = V1's PV1 is transitive

v1_pv2intr = V1's PV2 is intransitive

v1_pv2tr = V1's PV2 is transitive

v1_pv3intr = V1's PV3 is intransitive
v1_pv3tr = V1's PV3 is transitive

and for specification of arguments relative to each pv, the following holds:
The general pattern is using the full range of **Slot 3** labels prefixed by '**pv1**', '**pv2**' etc, as above, but now with an extra prefix indicating the **Vx** status in the sv; ex.:
v1_pv1suAg = in V1, the subject of PV1 is an Agent

$$\left[\mathrm{V1}\left[\mathrm{PV1}\left[\mathrm{GF}\left[\mathrm{SUBJ}\left[\mathrm{INDX}\left[\mathrm{ROLE\ agent}\right]\right]\right]\right]\right]\right]$$

(and likewise for all Vx, all PVx and all GFs and roles)
v1_pv1aspPerf = in V1, the aspect of PV1 is Perfective

Of particular relevance:
v1_pv2suClit = in V1, the subject of PV2 is realized as a cliticized pronoun
(and likewise for pv1, and pv3)

Identities across specific PVs:
v1_pv1suIDpv2su = inside V1, the subject of PV1 shares referent with subject of PV2

$$\left[\mathrm{V1}\left[\begin{array}{l}\mathrm{PV1}\left[\mathrm{GF}\left[\mathrm{SUBJ}\left[\mathrm{INDX}\boxed{1}\right]\right]\right]\\ \mathrm{PV2}\left[\mathrm{GF}\left[\mathrm{SUBJ}\left[\mathrm{INDX}\boxed{1}\right]\right]\right]\end{array}\right]\right]$$

v1_pv2aspIDpv3asp = inside V1, the aspect of PV2 is identical to the aspect of PV3

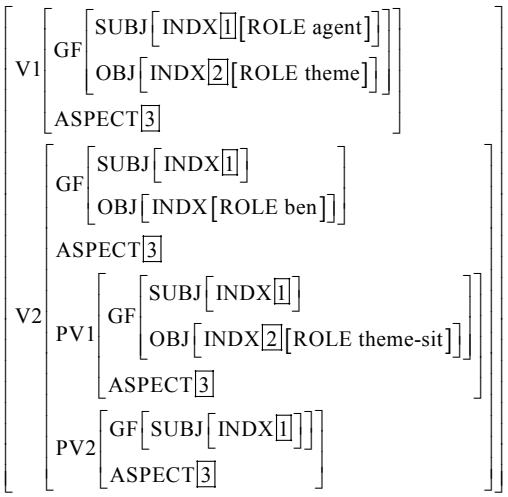
Frequently used specifications:
v2_ev2_suAspID
v2_pv1obThsit
v2_pv1obThsit
v2_pv2intr

An example of an ev as V2 in a serial verb construction, from Ga:

sv_suAspID_suAg-v1tr-v1obTh-v2_ev2_tr_suAspID -v2_pv1tr-v2_pv1obThsit-v2_pv2intr -v2tr-v2obBen

E-tao adeka kɛ-ba-ha mi
3S-search box move-come-give 1S
'He found a box for me.'

Explanation:
An sv with two verb constructions and sharing subject and aspect, with subjects being Agents; V1 is transitive and has a Theme object; V2 is an ev with two preverbs and being transitive; PV1 of V2 is transitive and the object of PV1 is a SitautionalTheme; PV2 of V2 is intransitive; the main verb is transitive and its object is a Beneficiary. Its AVM:



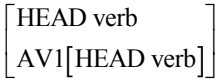
In this formula, the part v2_ev2_tr is redundant, since subsequent specifications indicate both that this is an ev (through 'pv') and that it is transitive. A slightly leaner version is thereby:
sv_suAspID_suAg-v1tr-v1obTh-v2_pv1tr-v2_pv1obThsit-v2_pv2intr-v2tr-v2obBen

B.2 Auxiliary Verb Constructions (AVCs/ axv)

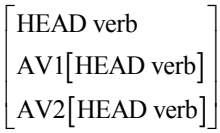
Under 'auxiliary verb' we subsume the *Modal*, *Perfective* and *Passive* auxiliaries of English, and counterparts of these in other languages. Like the preverbs of the EVCs, they cluster in a fixed order preceding the main verb, and our notation reflects this parallel between EVCs and AVCs. In logical structure, each pre-verb in an EVC typically relates to what follows in the way a V1 relates to V2 in an SVC, whereas in an AVC, the auxiliary is an operator with all the rest of the construction in its scope, thus like a main verb relative to its complement clause. Inflectionally, the TAM pattern inside an EVC is somewhat similar to that inside an SVC, whereas in an AVC, each auxiliary strictly governs the TAM of the verb following (like Modal requiring infinitive, Perfect requiring participle, etc.) The logical difference we assume to be included in the general definition of EVS vs AVC, i.e., in AREA 1 below, whereas the inflectional patterns can be indicated in AREA 2 specifications.

AREA 1 Global labels

axv = axv with one auxverb and the main verb

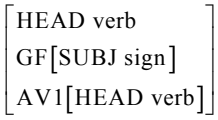


axv2 = axv with two auxverbs and the main verb

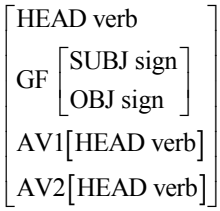


axv3 = axv with three auxverbs and the main verb

The following is a definition of the main verb inside an axv with regard to what valence it has, and thus the valence of the whole axv:
axv_intr = axv with one auxverb and an intransitive main verb

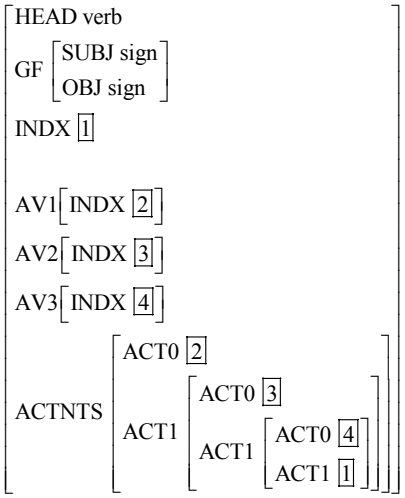


axv2_intr = axv with two auxverbs and an intransitive main verb
axv3_intr = axv with three auxverbs and an intransitive main verb
axv_tr = axv with one auxverb and a transitive main verb
axv2_tr = axv with two auxverbs and a transitive main verb



axv3_tr = axv with three auxverbs and a transitive main verb
axv_ditr = axv with one auxverb and a ditransitive main verb
axv2_ditr = axv with two auxverbs and a ditransitive main verb
axv3_ditr = axv with three auxverbs and a ditransitive main verb

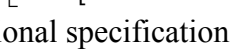
AVM displaying both syntax and semantics of **axv3_tr:**



AREA 2. **Specifications relative to each constituent auxverb**
Head category specification of the auxverb:
av1pass = auxverb 1 is passive (the auxverb of a periphrastic passive, like *be* in *be shot*)

av1perf = auxverb 1 is perfective (the auxverb of a periphrastic perfect, like *have* in *have seen*)

av1mod = auxverb 1 is modal



Inflectional specification of the auxverb:
av1tamPres = auxverb 1's inflection (for TAM) is Present tense

av1tamPtcpl = auxverb 1's inflection (for TAM) is (Perfect/Passive) Participle

[AV1 [HEAD[FORMATIVES ⟨ptcpl⟩]]]

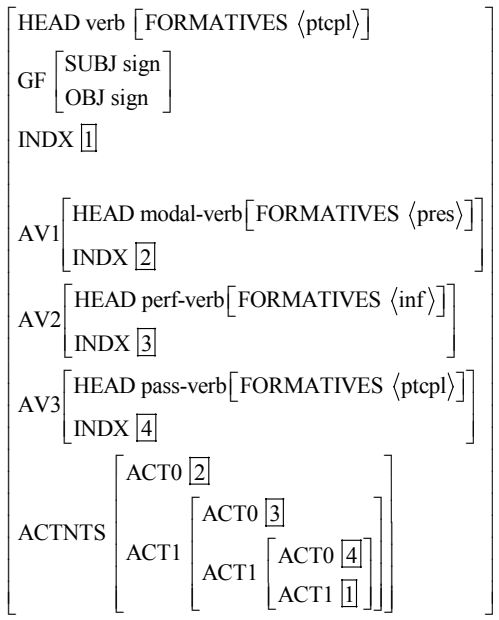
av1tamInf = auxverb 1's inflection (for TAM) is Infinitive

[AV1 [HEAD[FORMATIVES ⟨inf⟩]]]

Example:
axv_intrPs-av1mod-av2perf-av3pass
and
axv_intrPs-av1mod_av1tamPres-av2perf_av2tamInf-av3pass_av3tamPtcpl-vTamPtcpl
both describe the construction of the sentence

he may have been shot

The AVM induced by the longer template is:



C. Verbids

These are here regarded as something close to oblique constituents, but with verbal heads rather than prepositional heads. See definitions of the Slot 2-labels:
intrVid
intrVidScpr
trVid
trLghtVid
ditrVid

See definitions of the Slot 3 & 4 labels:
vidObLoc
vidObEndpt
vidObBPspec

V. Possible applications of the system, and discussion

A. Ordering of templates in an inventory

The following schema of ordering applies:

intr
tr
ditr
cop

For 'intr*X*', 'intr*Y*', where X, Y is 'scpr', 'adv', 'comp', etc., the order between 'intrX' and 'intrY' is alphabetical relative to X and Y. Likewise for 'tr', etc:

intr
intrX
intrY
tr
trX
trY
ditr
ditrX
ditrY
cop
copX
copY

For each of the above, templates where no item occurs in slot 3 go before templates *with* an item in slot 3.
When there are more than one item in slot 3, the linear precedence inside the slot is:

su > ob > iob > obl > comp > epon > sc > ..ID.. > ..

When template ordering is based on items occurring in slot 3, those with initial 'su' take precedence over others (no matter how long the sequence is), next those with intial 'ob' take precedence, etc., following the above precedence scheme. Likewise, when ordering is done according to what occurs in second position in slot 3, the same principles apply, and likewise for any further position.

The above principles form the core ordering. When templates are equal relative to those principles, templates with no labels in slot 4 precede templates *with* labels in slot 4. When there are more than one item in slot 4, the linear precedence inside slot 4 is keyed by the GF-initials, again by the precedence "su > ob > iob > obl > comp > epon > sc". When two templates are equal up to the slot 4 specification 'su…', then ranking is determined by

suAg > suCog > suSens > suExp

and correspondingly for roles relative to the other GFs.

Next on the priority list is slot 1: here plain 'v' goes before 'v_formative', and in the latter case, the fewer formatives go before the more formatives; precedence is otherwise alphabetical.
Last, as for slot 5, templates with no slot 5 item rank before templates with a slot 5 item. Among templates with a slot 5 item, precedence is alphabetical.

In a phase when one's main concern is to identify new constructions and templates, strict adherence to these ranking principles is of course not mandatory (and the list in section IV is a case in point), but the sooner one pays attention to them, the better.

B. Cross-linguistic uses of inventory lists

There are two main scenarios for cross-linguistic use of the lists: one when one establishes a first inventory for a language, and one for comparing established inventories:

For establishing an inventory for a new language, an already constructed list can serve as a **check-list**: In addressing Ewe, for instance, one can take departure in the list for Ga and go systematically down the list, judging for each Ga case whether there is a counterpart in Ewe. The range of full counterparts may give a substantive list already, and then near-counterparts can be characterized and filled into the list which then gradually gets 'customized' for Ewe. Not unlikely, as the Ewe list expands, cases may be found having counterparts in Ga although not yet on the Ga list; and so the lists expand interactively. At this stage of the process, it will matter that one knows precisely where in a given list a certain template would have its place, be this an exact point, or a span ("after this but before that" – it is like searching in a library shelf).

Comparison of established inventories may always have the dynamic bi-effect of enrichments incurring in one or both of the inventories; whether it does or not, **a strict common ordering facilitates the search for equivalents**, which can then be done by eye exclusively. String-search is always a fall-back strategy, but takes longer time.

C. Applying lists in Lexicography

From an inventory of single-verb constructions for a given language, the construction templates can be adapted as **types of lexical entries** for verbs of that language, reflecting the properties encoded as **subcategorization frames**. Into this inventory of contextual frames, the verbs heading example sentences in the templates list can be extracted as first members of the respective categories, and one can then systematically add verbs of the language to all the frames they can occur in.

With a complete inventory of construction types, and a complete inventory of verb lemmas of a language, one can establish which verbs employ a given construction type, and which construction types accommodate a given verb. A **verb class** can then identified as a set of verbs which are accommodated by the same set of construction types.

This notion of 'verb class' is related to that employed in (Levin 1993), which is based on *alternations* between construction types. An alternation, such as the 'spray-load alternation', can be viewed as a *pair* of construction types in which a number of verbs can participate, typically with rather similar semantics, highlighting – by a 'minimal pair' technique - semantic properties of the constructions chosen. For instance, the verb *load* can be used in sentences such as *He loaded hay onto the wagon* and *He loaded the wagon with hay*. Also *spray* can be used in this pair of construction types, and so *spray* and *load* can be categorized as belonging to the given class called the 'spray-load alternation'.

Joint membership in such a pair of constructions is by itself not a guarantee that the verbs in question have all occurrence frames in common; the latter is what is covered by the notion 'verb class' introduced above. For the situation where verbs have some, but not necessarily all frames in common, we may use the term **partial verb class**.

D. Establishing frequency of construction types

An inventory of construction types manifest in a language does not by itself give a full picture of what construction types 'pervade' the language (and thus give 'character' to it): to establish this, one also needs to establish the frequency of occurrence of the various types.
One way to accomplish this is through annotation of text with templates as here introduced, and then counting relevant occurrences. This does not require the existence of a lexicon attuned to the categories in question.
However, once one has an attuned lexicon, text search can also be made relative to occurrences of verbs, some entries of which correspond to the types in question. This will not require a previously annotated text, but will require a 'manual' check for each verb occurrence, that it actually occurs in the relevant frame.

E. The template system and its relation to grammars

A grammar for a language is a set of licencing conditions for constructions of the language, without displaying the licensed constructions themselves. The template system is, roughly speaking, orthogonal in function to that of a grammar; thus, they are supplementary to each other.
If cautiously designed, a template system should be able to communicate with a variety of grammar frameworks and formalisms, by virtue of employing notions recognized across theories and frameworks, and still having a recognizable accommodation within all of them. The labels outlined above are mildly oriented towards generative grammar, and to the extent some of them prove to be too parochial to this tradition, an interesting extension of the system will be the development of alternative labels addressing other traditions, but within definable equivalence or subsumption relations relative to the original labels.
In certain branches of grammar making, especially computational grammars, *test suites* are crucial in recording progress over time for the grammar development. Test suites are normally designed relative to each language (even each grammar), and often consist solely of the sentences themselves. An initiative started in the mid eighties (cf. Lehmann et al. 1996, Flickinger et al. 1987) was to somehow *index* test suites, to make them expose more explicitly what phenomena each sentence represents. The template system may be seen as a contribution to this enterprise, and hopefully so in a way suitable to grammars across frameworks.

The units with which the template system and grammars deal are in principle compatible, partly identical, and so one may explore how intimately the two approaches can be connected, while maintaining their distinct purposes. For instance, consider the correspondences (2) and (3) from above, repeated:

(2) v-tr-suAg_obAffincrem-COMPLETED_MONODEVMNT -

$$\begin{bmatrix} \text{HEAD verb} \\ \text{GF} \begin{bmatrix} \text{SUBJ} \left[\text{INDX } \boxed{1}\left[\text{ROLE agent}\right]\right] \\ \text{OBJ} \left[\text{INDX } \boxed{2}\left[\text{ROLE aff-increm}\right]\right] \end{bmatrix} \\ \text{INDX ref-index} \\ \text{ASPECT completed} \\ \text{ACTNTS} \begin{bmatrix} \text{ACT1 } \boxed{1} \\ \text{ACT2 } \boxed{2} \end{bmatrix} \\ \text{SIT-TYPE monotonic\_development} \end{bmatrix}$$

(3)

v - [HEAD verb]

tr - [GF [SUBJ [INDX 1]
OBJ [INDX 2]]
ACTNTS [ACT1 1
ACT2 2]]

suAg - [GF [SUBJ [INDX [ROLE agent]]]]

obAffincrem - [GF [OBJ [INDX [ROLE aff-increm]]]]

COMPLETED_MONODEVMNT - [ASPECT completed
SIT-TYPE monotonic_development]

One can well define a scenario where the AVMs to the right are actually produced in a parsing grammar, and where the correspondences to the labels on the left side are also integrated in this grammar. Hellan 2008a describes an architecture where this is possible, drawing on the correspondence that can be established between construction types and verb subcategorization frames (see V.C above). Assuming that lexical structures are substructures of structures generally provided by the grammar, an AVM like the one in (2) above can be associated with the verb *eat* as its lexical structure (aside from phonological and orthographic information, and more), and the template in (2) can be used as a **lexical type** for *eat*, connected to the AVM as its structural representation (e.g., as an LFG template, or an HPSG type). This lexical type, in turn, can be formally decomposed into its constituent parts with AVM definitions as suggested in (3) – for instance, in an HPSG/LKB setting, the following type definition could be stated composing the complex type *v-tr-suAg_obAffincrem-COMPLETED_MONODEVMNT*, with other type definitions accommodating the constituent labels (using the tdl style definitions used in LKB, cf. Copestake 2002, where ':=' means 'is a subtype of' and '&' expresses unification, thus implementing the 'merger' of the AVMs in (3) into the one AVM in (2)):

v-tr-suAg_obAffincrem-COMPLETED_MONODEVMNT :=
v & tr & suAg & obAffincrem & COMPLETED_MONODEVMNT

Such an exercise has been carried out for two HPSG grammars (for Norwegian and Ga) using the LKB system (Hellan 2007 and 2008b). It will seem that a similar conversion of labels could be done into an LFG grammar, and it might be interesting to explore whether it could be done for a GB/Minimalism type of grammar.

While this illustrates the principled possibility of integrating the template formalism and that of a grammar formalism, it by no means follows that for any given usage of the template formalism, there should exist a formal grammar reflecting the labels: on the contrary, this will rather be a rare situation. To restate our main point: the typological purpose of the template notation is to provide a compact way of representing an array of construction types hosted by a language, enabling efficient comparison, and still holding fairly detailed information expressed in a not too convoluted manner.

F. Is the template notation inherently restricted to verbal argument structure?

What is covered by the labeling system as presently given, is only a limited, although central aspect, of constructions of a language. Could the system be extended to cover other aspects of verbal constructions, such as modification, wh-movement, and more, and also constructions not headed by verbs? In principle any constructional domain where interesting information can be reached through attributes could be covered by the notational system – many labels in section II, for instance, make use of the attribute SPEC to expose properties of constituents of noun phrases. A general caveat is in order, however: the labeling system is not designed for taking over the role of constituent tree structures – for this, tree structures are far more elegant. Thus, in the context of analytic displays of sentence tokens, it would be wrong to try to encode all kinds of constituent properties into the string format. Likewise, the labeling format is not designed for taking over the role of standard morphological glossing.

Roughly speaking, one may say that for a somewhat complete morpho-syntactic and rudimentary semantic representation of a token sentence, a template representation of argument structure, a tree representation of constituent structure, and a standard morphological glossing, may serve together as a **representational triple** elucidating the different aspects of the sentence. This they may do without competing with possible *complete* representations of all these aspects as they might be devised in a full-fledged LFG or HPSG representation – both approaches are commendable, fullfilling different purposes (and the 'triple' approach may even serve in strategies of 'supertagging' tying the two approaches together – cf. Bangalore and Joshi 1999).

G. The template system as a construction ontology

The above-mentioned possibilities notwithstanding, the main role of the template system is that of a repository of free-standing representations of construction types. As such, the templates, as well as their constituent labels, might seem to lend themselves as possible items in an **ontology** of construction types. Since a full template is composed of information from different dimensions, such an ontology would have to be one using **multiple inheritance**. Moreover, since the information is complex, articulating it using **attributes and values** seems recommendable, with the possibility of attribute paths of length exceeding two, yielding AVMs like those exemplified above. A system equipped for an ontology with these properties is LKB (Copestake 2002), and a typed feature structure system which may count as an ontology has been developed using LKB for the labels listed throughout sections II and III, as well as the full templates listed in section IV and the template system for Norwegian referred to in section IV (Hellan 2008c).
(This type system is part of the grammar referred to in subsection E above, enabled by the fact that LKB allows parsing grammars to be constructed within taxonomies of linguistic objects.)

H. The template system as a shared methodology

Once a template system for a language or set of languages has been created, where – concretely – does it reside? With the creators, or in text file copies distributed among interested parties? In such a case, how are updates and improvements, and systems for new languages, integrated with the previosly existing versions?
One possibility may be to have generally accessible servers with Version Control systems, which merge existing material with new material and record the development from version to

version, and allow people to update their own versions from the Version Control, and check out versions for the first time.
Another possibility will be to have a wiki where people can post their contributions on line, and in addition have a discussion forum. Such a format is provided, e.g., by typecraft.org, which also has an interface for producing the glossing and template parts of the representational triples mentioned in subsection F above.
Combinations of these may also be possible: what is clear is that the template system is most profitably conducted as a **shared methodology**, of which both approaches mentioned are examples.

References

Bangalore, Srinivas, and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics* 25(2):237-265

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation*, Coling 2002, Taipei.

Butt, Miriam, Tracy Holloway King, Maria-Eugenia Nini and Frederique Segond. 1999. *A Grammar-writer's Cookbook*. Stanford: CSLI Publications.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars.* CSLI Publications, Stanford.

Dakubu, Mary E. K. 2008. The Construction label project: a tool for typological study. Presented at West African Languages Congress (WALC), Winneba, July 2008.

Flickinger, Daniel, John Nerbonne, Ivan A. Sag, and Thomas Wassow. 1987. Toward Evaluation of NLP Systems. Technical report. Hewlett-Packard Laboratories. Distributed at the 24th Annual Meeting of the Association for Computational Linguistics (ACL).

Hellan, Lars. 2007. On 'Deep Evaluation' for Individual Computational Grammars and for Cross-Framework Comparison. In: T.H. King and E. M. Bender (eds) *Proceedings of the GEAF 2007 Workshop*. CSLI Studies in Computational Linguistics ONLINE. CSLI Publications. *http://csli-publications.stanford.edu/*

Hellan, Lars. 2008a. Enumerating Verb Constructions Cross-linguistically. COLING Workshop on Grammar Engineering Across frameworks. Manchester.

Hellan, Lars. 2008b. A package of phenomena-oriented LKB grammars. Class material, NTNU

Hellan, Lars. 2008c. An LKB system for verb construction types. NTNU

Hellan, Lars. 2009. A basic introduction to using attribute-value matrices (AVMs) in linguistic representation.

Hellan, Lars., Lars Johnsen and Anneliese Pitz. 1989. TROLL. Ms., NTNU

Lehmann, Sabine., S. Oepen, S. Regier-Prost, K. Netter, V. Lux, J. Klein, K. Falkedal, F. Fouvry, D. Estival, E. Dauphin, H. Compagnion ,J. Baur, L. Balkan, D. Arnold. 1996. Test Suites for Natural Language Processing. *Proceedings of COLING* 16, p. 711-16.

Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.