DERIVATION in the CONSTRUCTION LABELING SYSTEM

Continuing from:

Studies in the Languages of the Volta Basin 6. Part 3
IDENTIFYING VERB CONSTRUCTIONS CROSS-LINGUISTICALLY, by
Lars Hellan, NTNU, and Mary Esther Kropp Dakubu, University of Ghana

Derived valencies

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 end of section IIc for a corresponding annotation relative to the grammatical functions involved.

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)

Derivational (operational) specifications of constituents
These specifications trace the derivational history of a GF, in a way similar to 'chains' in GB and Relational Grammar.

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

Effects of iteration of derivation (one operation having applied to the output of another):
`suPobCsu` = **su** which would have been *ob* relative to *input* of *Passive* formation, where this ob would have been *su* relative to *input* of *Causative* formation
`suPobCob` = **su** which would have been *ob* relative to *input* of *Passive* formation, where this ob would have been *ob* relative to *input* of *Causative* formation
`suPobCob2` = **su** which would have been *ob* relative to *input* of *Passive* formation, where this ob would have been *ob2* relative to *input* of *Causative* formation
`suPobCiob` = **su** which would have been *ob* relative to *input* of *Passive* formation, where this ob would have been *iob* relative to *input* of *Causative* formation
`suPobCobl` = **su** which would have been *ob* relative to *input* of *Passive* formation, where this ob would have been *obl* relative to *input* of *Causative* formation

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

`suPiobCsu` = **su** which would have been i*ob* relative to *input* of *Passive* formation, where this iob would have been *su* relative to *input* of *Causative* formation
`suPiobCob` = **su** which would have been i*ob* relative to *input* of *Passive* formation, where this iob would have been *ob* relative to *input* of *Causative* formation
`suPiobCob2` = **su** which would have been i*ob* relative to *input* of *Passive* formation, where this iob would have been *ob2* relative to *input* of *Causative* formation
`suPiobCiob` = **su** which would have been i*ob* relative to *input* of *Passive* formation, where this iob would have been *iob* relative to *input* of *Causative* formation
`suPiobCobl` = **su** which would have been i*ob* relative to *input* of *Passive* formation, where this iob would have been *obl* relative to *input* of *Causative* formation

`suPoblCsu` = **su** which would have been *obl* relative to *input* of *Passive* formation, where this obl would have been *su* relative to *input* of *Causative* formation
`suPoblCob` = **su** which would have been *obl* relative to *input* of *Passive* formation, where this obl would have been *ob* relative to *input* of *Causative* formation
`suPoblCob2` = **su** which would have been *obl* relative to *input* of *Passive* formation, where this obl would have been *ob2* relative to *input* of *Causative* formation
`suPoblCiob` = **su** which would have been *obl* relative to *input* of *Passive* formation, where this obl would have been *iob* relative to *input* of *Causative* formation
`suPoblCobl` = **su** which would have been *obl* relative to *input* of *Passive* formation, where this obl would have been *obl* relative to *input* of *Causative* formation
`suPobAobl` = **su** which would have been *ob* relative to *input* of *Passive* formation, where this ob would have been *obl* relative to *input* of *Applicative* formation
`suPob2Aobl` = **su** which would have been *ob2* relative to *input* of *Passive* formation, where this ob2 would have been *obl* relative to *input* of *Applicative* formation
`suPiobAobl` = **su** which would have been i*ob* relative to *input* of *Passive* formation, where this iob 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