Lexical Mapping Theory and the anatomy of a (verbal) lexical entry

Jamie Y. Findlay
jamie.findlay@ling-phil.ox.ac.uk

University of Oxford

25th International Lexical Functional Grammar Conference
(Oslo/Online)

Outline

1. Motivations and goals
2. The anatomy of a lexical entry
   - Core meaning
   - Valency frame(s)
   - Mapping principles
   - Argument alternant(s)
3. Conclusions

Motivations and goals

LMT

- **(Lexical) Mapping Theory (LMT):** a theory of the linking between semantic arguments and grammatical functions.
  (e.g. Bresnan & Kanerva 1989; Kibort 2007)
  - Some recent work has been skeptical of the need for an independent level of a-structure over which LMT is to operate.
    (Asudeh & Giorgolo 2012; Asudeh et al. 2014; Findlay 2016)

Motivations

- A continuation of the research programme started by Asudeh & Giorgolo (2012).
  - A desire for ontological parsimony: no need for a-structure.
  - An uneasiness with the formal underpinnings of LMT.
  - A drive to modularity in the lexicon, using templates.

Tools of the trade

- LMT introduces a number of new formal tools into the LFG architecture, the consequences of which are sometimes not made clear, or are dismissed:
  - Feature decomposition:

	$-r$	$+r$
$-o$	SUBJ	OBL_θ
$+o$	OBJ	OBJ_θ

    - ★ GFs are no longer theoretical primitives (Butt 1995: 31).
    - ★ What is the status of these features? (See brief discussion in Findlay 2016: 298–299.)
  - ‘Pre-lexical derivation sequences’:
    - ★ Where does mapping fit into the LFG parsing algorithm?
    - ★ By what mechanisms does it operate?

Tools of the trade

- Sometimes formal extensions are necessary, but . . .
  - All things being equal, we prefer sparser theories.
  - If we do add extra tools, their formal/mathematical properties must be clear.

Goal 1

Show that the insights of LMT can be expressed using existing LFG machinery.

A modular lexicon

- One strand of research in LFG has also advocated a highly modular view of the lexicon: (e.g. Asudeh & Giorgolo 2012; Asudeh et al. 2013; Przepiórkowski 2017)
  - Lexical entries consist of an idiosyncratic core, containing e.g. lexical meaning, supplemented *monotonically* by additional information.
  - This information is represented in **templates** (Dalrymple et al. 2004) which capture cross-lexical commonalities.
    (Cf. Przepiórkowski 2017 for a well-developed version of this view.)
- One major advantage of this view is that it enables us to represent information at a higher level of abstraction, packaging up the underlying implementation and leaving only the theoretically interesting facts.

Goal 2

(a) Break down a lexical entry into identifiable parts.

(b) Factor out the contents of these parts so that they can be described using high-level templates.

The anatomy of a lexical entry

Components of a (verbal) lexical entry

(1) form, category;
functional description:
- core meaning
- valency frame(s)
- argument alternation(s)
- other material

Core meaning

Core meaning

(2) gave, V;
$(\uparrow \text{PRED}) = \text{`give'}$
$(\uparrow_\sigma \text{REL}) = \text{give}$

$\lambda x \lambda y \lambda z \lambda e.\mathbf{give}(e) \wedge \mathbf{agent}(e, x) \wedge \mathbf{theme}(e, y) \wedge \mathbf{beneficiary}(e, z)$:
$(\uparrow_\sigma \text{ARG1}) \multimap (\uparrow_\sigma \text{ARG2}) \multimap (\uparrow_\sigma \text{ARG3}) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

$(\uparrow_\sigma \text{ARG1})$
$(\uparrow_\sigma \text{ARG2})$
$(\uparrow_\sigma \text{ARG3})$

Ensuring arguments are mapped

- The existential constraints mentioning the various ARGs require that *some* information about them is specified elsewhere.
- Assuming that nothing does so directly, this will ensure they must be mapped to a GF that can provide some information.

(3) Kim yawns.

(4) Kim, N;
$(\uparrow_\sigma \text{ REL}) = \text{Kim}$

(5) $(\uparrow \text{ SUBJ})_\sigma = (\uparrow_\sigma \text{ ARG}1)$

- Argument-suppressing operations will provide a dummy REL value, ‘var’ (for ‘variable’).

Valency frame(s)

Valency frames

- A verb is associated with one or more valency frames, which represent its arguments and their potential syntactic realisation.
- Such frames are equivalent to Kibort's (2001; 2007; 2008) a-structures.

(6) $\left\langle \begin{array}{cccccc} \text{arg}_1 & \text{arg}_2 & \text{arg}_3 & \text{arg}_4 & \ldots & \text{arg}_n \\ [-o]/[-r] & [-r] & [+o] & [-o] & & [-o] \end{array} \right\rangle$

(7) $\text{SUBJ} > \text{OBJ}, \text{OBL}_\theta > \text{OBJ}_\theta$

- What Kibort's valency frame + Mapping Principle really give us is a *default* mapping for each arg position, plus a possible alternative.
- But this is provided via *sui generis* a-structure and a separate mapping algorithm.
- We can achieve the same result using standard LFG functional descriptions.

Some basic templates

(8) a. $\textsc{Map}(X, Y) :=$
$(\uparrow X)_\sigma = (\uparrow_\sigma\ Y)$

b. $@\textsc{Map}(\text{SUBJ, ARG1}) \equiv$
$(\uparrow \text{SUBJ})_\sigma = (\uparrow_\sigma\ \text{ARG1})$

(9) $\textsc{NoMap}(Y) :=$
$(\uparrow_\sigma\ Y)_{\sigma^{-1}} = \varnothing$

(10) a. $\text{MINUSO} \equiv \{\text{SUBJ}|\text{OBL}_\theta\}$

b. $\text{PLUSO} \equiv \{\text{OBJ}|\text{OBJ}_\theta\}$

c. $\text{MINUSR} \equiv \{\text{SUBJ}|\text{OBJ}\}$

d. $\text{PLUSR} \equiv \{\text{OBJ}_\theta|\text{OBL}_\theta\}$

Recasting Kibort's valency positions

(11) $\text{DEFAULT-SUBJECT-UNERG}(\textit{arg}) \coloneqq$
$\{ \text{@MAP}(\text{SUBJ}, \textit{arg}) \mid \neg\text{@MAP}(\text{SUBJ}, \textit{arg}) \wedge \neg\text{@MAP}(\text{PLUSO}, \textit{arg}) \}$
%arg1 = *arg*

- One positive specification, one set of negative specifications.
  - With no further information, the first disjunct must be true, since the existential equations in the core require *some* positive specification of the mapping between GFs and ARGs.
- Local name assigned to the argument, intended to be mnemonic for the arg positions in Kibort's theory.
  - This ensures that further mapping rules apply to the correct argument, without needing to imbue s-structure labels with meaning.

(12) a. DEFAULT-SUBJECT-UNACC(*arg*) :=
$\{$@MAP(SUBJ, *arg*) | ¬@MAP(SUBJ, *arg*) ∧ ¬@MAP(PLUSR, *arg*)$\}$
%arg1 = *arg*

b. DEFAULT-OBJECT(*arg*) :=
$\{$@MAP(OBJ, *arg*) | ¬@MAP(OBJ, *arg*) ∧ ¬@MAP(PLUSR, *arg*)$\}$
@MAPPING-PRINCIPLE-ARG2
%arg2 = *arg*

c. DEFAULT-OBJTH(*arg*) :=
$\{$@MAP(OBJ_θ, *arg*) | ¬@MAP(OBJ_θ, *arg*) ∧ ¬@MAP(MINUSO, *arg*)$\}$
@MAPPING-PRINCIPLE-ARG3
%arg3 = *arg*

d. DEFAULT-OBL(*arg*) :=
$\{$@MAP(OBL_θ, *arg*) | ¬@MAP(OBL_θ, *arg*) ∧ ¬@MAP(PLUSO, *arg*)$\}$
@MAPPING-PRINCIPLE-ARG4
%arg4 = *arg*

Love

- Meaning:

(13) $\lambda x \lambda y \lambda e.\mathbf{love}(e) \wedge \mathbf{agent}(e) = x \wedge \mathbf{theme}(e) = y :$
$(\uparrow_\sigma \text{ ARG1}) \multimap (\uparrow_\sigma \text{ ARG2}) \multimap (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$

- A-structure:

(14)

	ARG1 (agent)	ARG2 (theme)	
$\langle$	arg_1	arg_2	$\rangle$
	$[-o]$	$[-r]$	

- Templatic valency frame:

(15) a. CANONICAL-TRANSITIVE(X, Y) :=
@DEFAULT-SUBJECT-UNERG(X)
@DEFAULT-OBJECT(Y)

b. @CANONICAL-TRANSITIVE(ARG1, ARG2)

Give

- A verb like *give* has two different valency frames:

(16) a. Non-dative-shifted: *Odo gave a gift to Kira.*

b. Dative-shifted: *Odo gave Kira a gift.*

(17) a.

	ARG1 (agent)	ARG2 (theme)	ARG3 (beneficiary)	
⟨	arg_1	arg_2	arg_4	⟩
	[$-o$]	[$-r$]	[$-o$]	

b.

	ARG1 (agent)	ARG3 (beneficiary)	ARG2 (theme)	
⟨	arg_1	arg_2	arg_3	⟩
	[$-o$]	[$-r$]	[$+o$]	

Give

Odo gave a gift to Kira.

	ARG1 (agent)	ARG2 (theme)	ARG3 (beneficiary)	
⟨	arg_1	arg_2	arg_4	⟩
	$[-o]$	$[-r]$	$[-o]$	

Odo gave Kira a gift.

	ARG1 (agent)	ARG3 (beneficiary)	ARG2 (theme)	
⟨	arg_1	arg_2	arg_3	⟩
	$[-o]$	$[-r]$	$[+o]$	

(18) @DEFAULT-SUBJECT-UNERG(ARG1)

(19) a. OBL-BEN(*X*, *Y*) :=
@DEFAULT-OBJECT(*X*)
@DEFAULT-OBL(*Y*)

b. @OBL-BEN(ARG2, ARG3)

(20) a. OBJ-BEN(*X*, *Y*) :=
@DEFAULT-OBJTH(*X*)
@DEFAULT-OBJECT(*Y*)

b. @OBJ-BEN(ARG2, ARG3)

Give

- The two possibilities can be captured in a higher-level template:

(21) EN-DITRANSITIVE(X, Y, Z) :=
@DEFAULT-SUBJ-UNERG(X)
$\{$@OBL-BEN(Y, Z) $|$ @OBJ-BEN(Y, Z)$\}$

Mapping principles

Mapping principles

- The mechanism by which GFs and ARGs are aligned, called the Mapping Principle by Kibort, is a little mysterious formally speaking.
  - Often some loose appeal to OT or a similar process (e.g. Butt et al. 1997, Findlay 2016: 322), or once again to a *sui generis* mechanism.
- But its effects can be captured using standard LFG functional descriptions – this is what the various MAPPING-PRINCIPLE templates do.
- Each position below arg_1 in Kibort's theory is essentially in competition with some higher arg position:
  - arg_2 wants to be a SUBJ, but is generally blocked from doing do by arg_1
  - arg_3 wants to be an OBJ, but is generally blocked from doing do by arg_2
  - arg_4 wants to be a SUBJ, but is generally blocked from doing do by arg_1

Mapping principles

- We can capture this in a disjunction:

(22) $\textsc{Mapping-Principle-Arg2} \coloneqq$

$$\left\{\begin{matrix} (\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \text{ \%arg2}) \\ \vee \\ (\uparrow \textsc{subj}) \\ (\uparrow \textsc{subj})_\sigma \neq (\uparrow_\sigma \text{ \%arg2}) \\ \vee \\ @\textsc{NoMap}(\text{\%arg2}) \end{matrix}\right\}$$

- If SUBJ maps to %arg1, the first disjunct here cannot be true, since it would make σ non-functional.
- The second disjunct provides another negative constraint on %arg2, which forces the default OBJ mapping, all other things being equal.
  - The existential constraint in this disjunct ensures that the default mapping only applies when something else fills %arg2's preferred GF. If nothing else realises SUBJ, then %arg2 will.

Mapping principles

(23) $\textsc{Mapping-Principle-Arg3} \coloneqq$

$$\left\{\begin{matrix} (\uparrow \text{OBJ})_\sigma = (\uparrow_\sigma \%\text{arg3}) \\ \vee \\ (\uparrow \text{OBJ}) \\ (\uparrow \text{OBJ})_\sigma \neq (\uparrow_\sigma \%\text{arg3}) \\ \vee \\ @\textsc{NoMap}(\%\text{arg3}) \end{matrix}\right\}$$

(24) $\textsc{Mapping-Principle-Arg4} \coloneqq$

$$\left\{\begin{matrix} (\uparrow \text{SUBJ})_\sigma = (\uparrow_\sigma \%\text{arg4}) \\ \vee \\ (\uparrow \text{SUBJ}) \\ (\uparrow \text{SUBJ})_\sigma \neq (\uparrow_\sigma \%\text{arg4}) \\ \vee \\ @\textsc{NoMap}(\%\text{arg4}) \end{matrix}\right\}$$

Argument alternant(s)

Argument alternations

- With the valency frames in place, we can model argument alternations in the same way as in Kibort's LMT, as the further specification of a particular argument's GF.

  (As cashed out in e.g. Asudeh et al. 2014 and Findlay 2016.)

(25) Chichewâ locative inversion (Bresnan & Kanerva 1989: 2):

a. Chi-tsîme chi-li ku-mu-dzi. (SUBJ, OBL_{LOC})
*7-well 7*SUBJ*-be 17-3-village*
'The well is in the village.'

b. Ku-mu-dzi ku-li chi-tsîme. (SUBJ, OBJ)
*17-3-village 17*SUBJ*-be 7-well*
'In the village is a well.'

Argument alternations

(26) LOCATIVE-INVERSION :=
@MAP(PLUSO, %arg1)
(after Kibort 2007)

(27) DEFAULT-SUBJECT-UNACC(*arg*) :=
$\{$@MAP(SUBJ, *arg*) $|$ $\neg$@MAP(SUBJ, *arg*) $\wedge$ $\neg$@MAP(PLUSR, *arg*)$\}$
%arg1 = *arg*

(28) MAPPING-PRINCIPLE-ARG4 :=

$$\left\{ \begin{array}{c} (\uparrow \text{SUBJ})_\sigma = (\uparrow_\sigma \text{ \%arg4}) \\ \vee \\ (\uparrow \text{SUBJ}) \\ (\uparrow \text{SUBJ})_\sigma \neq (\uparrow_\sigma \text{ \%arg4}) \\ \vee \\ \text{@NoMap(\%arg4)} \end{array} \right\}$$

Argument suppression

- Argument suppressing alternations are different, since we also need to do something about the argument which isn't overtly realised.

(29) SUPPRESS(*arg, template*) :=
@NOMAP(*arg*)
@*template*(*arg*)
($\uparrow_\sigma$ *arg* REL) = var

CLOSE-OFF

(30) CLOSE-OFF(*arg*) :=
$\lambda P.\exists x[P(x)] : [(\uparrow_\sigma\ arg) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$

(31) a. Jadzia was killed.

b. $\exists e \exists x[\mathbf{kill}(e) \wedge \mathbf{agent}(e) = x \wedge \mathbf{patient}(e) = \mathbf{jadzia}]$

(32) PASSIVE :=
(↑ VOICE) = PASSIVE
{@SHORT-PASSIVE | @LONG-PASSIVE}

(33) a. SHORT-PASSIVE :=
@SUPPRESS(%arg1, CLOSE-OFF)

b. LONG-PASSIVE :=
@MAP(PLUSR, %arg1)

(34) ACTIVE :=
(↑ VOICE) = ACTIVE

CLOSE-OFF

(35) French middle voice (Grimshaw 1982)

a. La librairie vend beaucoup de livres.
the bookshop sells many books
'The bookshop sells many books.'

b. Beaucoup de livres se vendent dans cette ville.
many books SE *sell in this town*
'Many books are sold in this town.'

(36) se, CL;
(↑ REFL) = +
(↑ SUBJ PERS) = 3

(37) FR-MIDDLE :=
@SUPPRESS(%arg1, CLOSE-OFF)
(↑ REFL) $=_c$ +

BIND

(Grimshaw 1982)

(38) Jean se voit.
John SE *sees*
'John sees himself.'

(39) a. Un train passe toutes les heures.
a train passes all the hours
'A train goes by every hour.'

b. Il passe un train toutes les heures.

(40) a. Un train conduira les voyageurs à Paris.
a train will.take the passengers to Paris
'A train will take the travellers to Paris.'

b. *Il conduira un train les voyageurs à Paris.

c. *Il conduira les voyageurs un train à Paris.

BIND

(41) a. Une femme s' est offerte pour mener le combat.
a woman SE AUX *offered to lead the combat*
'A woman offered herself to lead the fighting.'

b. Il s'est offert une femme pour mener le combat.

(42) BIND(arg_β, arg_α) ≔
$\lambda P \lambda x.P(x)(x) : [(\uparrow_\sigma\ arg_\alpha) \multimap (\uparrow_\sigma\ arg_\beta) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma\ arg_\beta) \multimap \uparrow_\sigma$

(43) FR-REFLEXIVE ≔
@SUPPRESS(%arg2, BIND(%arg1))
$(\uparrow \text{REFL}) =_c +$

Conclusions

A lexical entry

(44) given, V;
(↑ PRED) = 'give'
$(\uparrow_\sigma$ REL) = give

$\lambda x \lambda y \lambda z \lambda e.\textbf{give}(e) \wedge \textbf{agent}(e,x) \wedge \textbf{theme}(e,y) \wedge \textbf{beneficiary}(e,z)$:
$(\uparrow_\sigma \text{ ARG1}) \multimap (\uparrow_\sigma \text{ ARG2}) \multimap (\uparrow_\sigma \text{ ARG3}) \multimap (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$

$(\uparrow_\sigma \text{ ARG1})$
$(\uparrow_\sigma \text{ ARG2})$
$(\uparrow_\sigma \text{ ARG3})$

@EN-DITRANSITIVE (ARG1, ARG2, ARG3)

@PASSIVE

Conclusions

Goal 1

Show that the insights of LMT can be expressed using existing LFG machinery.

- The default GF and MAPPING-PRINCIPLE templates make this possible.
- The default templates assign a local name to the s-structure arguments, allowing us to straightforwardly simulate anything Kibort's LMT can say about monotonic manipulations of argument mappings.

Conclusions

Goal 2

(a) Break down a lexical entry into identifiable parts.

(b) Factor out the contents of these parts so that they can be described using high-level templates.

- A lexical entry contains at least
  - a core meaning, and
  - a valency frame (its ‘argument structure’),
- It may also contain
  - information about argument alternations
  - further lexically idiosyncratic information
- Information about tense, aspect, mood, etc. will also need to be integrated.

Conclusions

- By using templates, we conceal the underlying mathematical gore, but leave the theoretically interesting generalisations available for discussion and use.
- These proposals also offer a firmer formal foundation for insights gained by work couched in LMT, hopefully further promoting the profitable cross-fertilisation between Glue practitioners and those working on mapping theory which has flourished in the past 8+ years.

References

Asudeh, Ash, Mary Dalrymple & Ida Toivonen. 2013. Constructions with Lexical Integrity. *Journal of Language Modelling* 1(1). 1–54. http://jlm.ipipan.waw.pl/index.php/JLM/article/view/56/49.

Asudeh, Ash & Gianluca Giorgolo. 2012. Flexible composition for optional and derived arguments. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG12 Conference*, 64–84. Stanford, CA: CSLI Publications. http://www.stanford.edu/group/cslipublications/cslipublications/LFG/17/papers/lfg12asudehgiorgolo.pdf.

Asudeh, Ash, Gianluca Giorgolo & Ida Toivonen. 2014. Meaning and valency. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG14 Conference*, 68–88. CSLI Publications. http://web.stanford.edu/group/cslipublications/cslipublications/LFG/19/papers/lfg14asudehetal.pdf.

Bresnan, Joan & Jonni M. Kanerva. 1989. Locative inversion in Chichewa: A case study of factorization in grammar. *Linguistic Inquiry* 20(1). 1–50. https://www.jstor.org/stable/4178612.

Butt, Miriam. 1995. *The structure of complex predicates in Urdu*. Stanford, CA: CSLI Publications.

Butt, Miriam, Mary Dalrymple & Anette Frank. 1997. An architecture for linking theory in LFG. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG97 Conference*, Stanford, CA: CSLI Publications. http://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG2-1997/lfg97butt-dalrymple-frank.pdf.

Dalrymple, Mary, Ronald M. Kaplan & Tracy Holloway King. 2004. Linguistic generalizations over descriptions. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG04 Conference*, 199–208. Stanford, CA: CSLI Publications. https://web.stanford.edu/group/cslipublications/cslipublications/LFG/9/pdfs/lfg04dkk.pdf.

Findlay, Jamie Y. 2016. Mapping theory without argument structure. *Journal of Language Modelling* 4(2). 293–338. http://jlm.ipipan.waw.pl/index.php/JLM/article/view/171/148.

Grimshaw, Jane. 1982. On the lexical representation of Romance reflexive clitics. In Joan Bresnan (ed.), *The mental representation of grammatical relations*, 87–148. Cambridge, MA: MIT Press.

Kibort, Anna. 2001. The Polish passive and impersonal in Lexical Mapping Theory. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG01 Conference*, 163–183. Stanford, CA: CSLI Publications. https://web.stanford.edu/group/cslipublications/cslipublications/LFG/6/pdfs/lfg01kibort.pdf.

Kibort, Anna. 2007. Extending the applicability of Lexical Mapping Theory. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG07 Conference*, 250–270. Stanford, CA: CSLI Publications. http://www.stanford.edu/group/cslipublications/cslipublications/LFG/12/papers/lfg07kibort.pdf.

Kibort, Anna. 2008. On the syntax of ditransitive constructions. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG08 Conference*, 312–332. Stanford, CA: CSLI Publications. http://www.stanford.edu/group/cslipublications/cslipublications/LFG/13/papers/lfg08kibort.pdf.

Przepiórkowski, Adam. 2017. A full-fledged hierarchical lexicon in LFG: the FrameNet approach. In Victoria Rosén & Koenraad De Smedt (eds.), *The very model of a modern linguist: in honor of Helge Dyvik* (Bergen Language and Linguistics Studies 8), 202–219. Bergen, NO: University of Bergen. https://doi.org/10.15845/bells.v8i1.1336.

Simplifying the core

- We can further simplify the lexical representation of the core by factoring out shared aspects of the meaning.
  (Cf. e.g. Asudeh & Giorgolo 2012; Findlay 2016; Przepiórkowski 2017.)

- In this case, we can move them to a valency frame.

(45) EN-DITRANSITIVE $(X, Y, Z) :=$

$\{$@OBL-BEN$(X, Y, Z) \mid$ @OBJ-BEN$(X, Y, Z)\}$

$$\lambda P \lambda x \lambda y \lambda z \lambda e.P(e) \wedge \textbf{agent}(e,x) \wedge \textbf{theme}(e,y) \wedge \textbf{beneficiary}(e,z) :$$
$$[(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap$$
$$(\uparrow_\sigma X) \multimap (\uparrow_\sigma Y) \multimap (\uparrow_\sigma Z) \multimap (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$$

$(\uparrow_\sigma X)$
$(\uparrow_\sigma Y)$
$(\uparrow_\sigma Z)$

Simplifying the core

- Assuming that REL and PRED will always have the same value (*modulo* concerns about unique instantiation), we can also introduce a template LEXEME:

(46) LEXEME(X) :=
$(\uparrow \text{PRED}) = \text{`}X\text{'}$
$(\uparrow_{\sigma} \text{REL}) = X$

A lexical entry

(47) given, V;
@LEXEME(give)
$\lambda e.\textbf{give}(e) : (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$
@EN-DITRANSITIVE (ARG1, ARG2, ARG3)
@PASSIVE

A note on EVENT

- In many approaches to event semantics in Glue, an s-structure attribute EVENT is assumed, but no positive defining equation for it is supplied.
- Doesn't this cause a problem for a meaning constructor like (48)?

(48) $\lambda x \lambda y \lambda e.\mathbf{love}(e) \wedge \mathbf{agent}(e) = x \wedge \mathbf{theme}(e) = y :$
$(\uparrow_\sigma \text{ ARG1}) \multimap (\uparrow_\sigma \text{ ARG2}) \multimap (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$

- In fact, no: the resource $(\uparrow_\sigma \text{ EVENT})$ is never *directly* consumed. For example, once the verb has combined with its arguments, a meaning constructor like (49) will come along and consume the remaining dependency on $(\uparrow_\sigma \text{ EVENT})$:

(49) $\lambda P.\exists e[P(e) \wedge \mathbf{past}(e)] : [(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$

- The whole conditional statement is consumed, not $(\uparrow_\sigma \text{ EVENT})$ alone.
- An expression like $(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$ says '*If* you had a resource $(\uparrow_\sigma \text{ EVENT})$, then you could produce a resource $\uparrow_\sigma$', but such a condition is never met, and so the resource remains purely hypothetical.
- The actual error in previous Glue work was including EVENT in the semantic structures. Such an attribute doesn't exist, and doesn't need to – it is a hypothetical resource.