

# The LOGON MT infrastructure redux

## Transfer rules in more detail

---

Ling 567

March 3, 2026

# Overview

---

- MRS and MT: Some history (reprise)
- The LOGON architecture
  - Processing steps
  - Transfer rules
  - VPM
- Stepping through our MMT sentences

# MRS and MT: Some history

---

- Copestake et al 1995: Original motivation for MRS included MT applications
- Resolving scope ambiguities is hard, and usually not necessary
  - Logical form equivalence is undecidable even in FOPL (Shieber 1993)
- Mimicking syntactic structure in semantics makes transfer harder
  - *fierce black cat* <> *gato negro y feroz* (Spanish)
  - *young black bull* <> *novillo negro*
- MRS gives logical forms with less syntactic complexity and underspecification wherever possible.

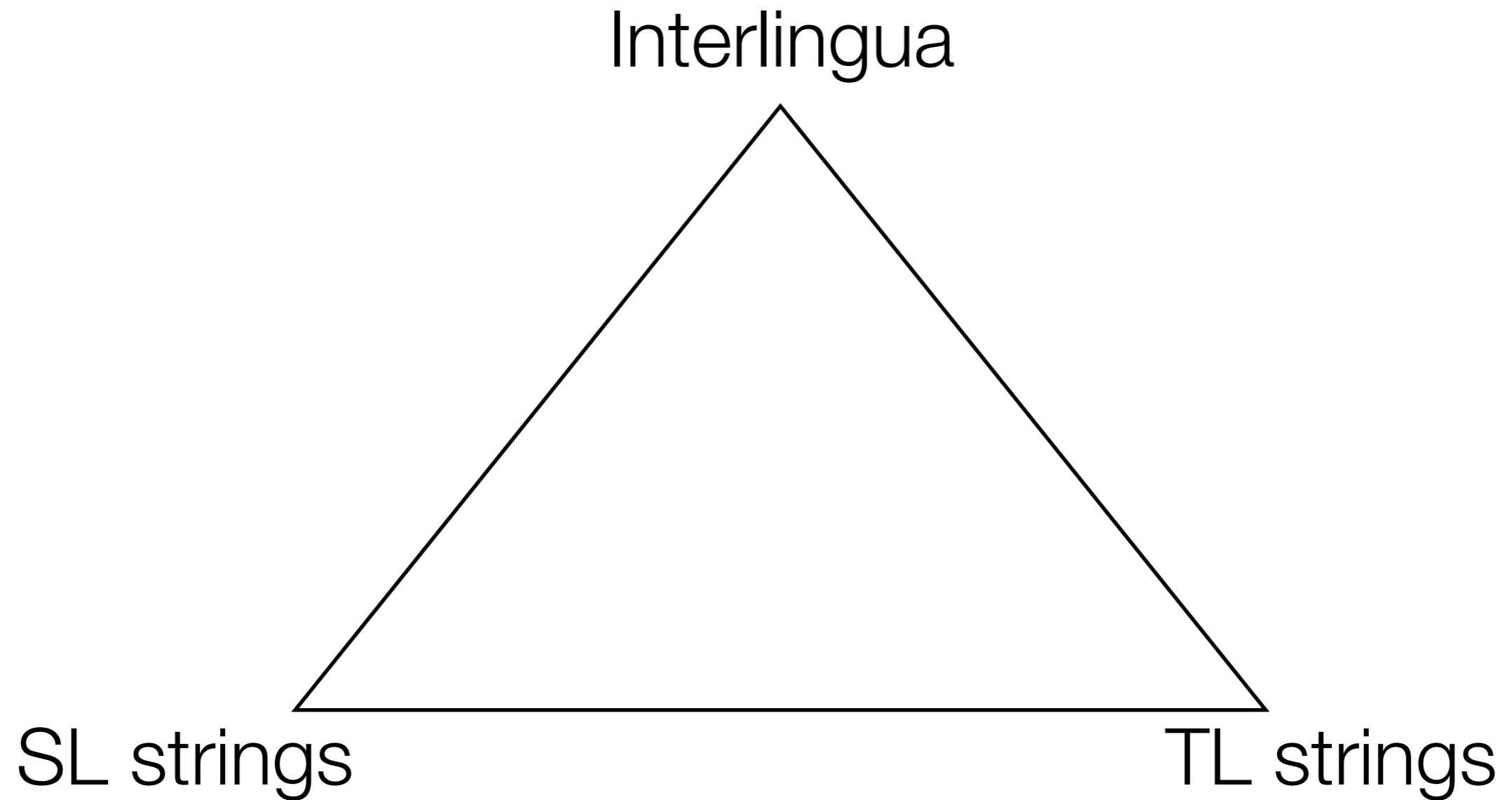
# MRS and MT: Some history

---

- MRS originally developed in the context of *VerbMobil* but not fully deployed for transfer-based MT in that project.
- In 2003, LOGON (Oepen et al 2004, 2007) up the thread and builds the first MRS-based MT system. (Norwegian -> English; tourism brochures)
- Input is LFG, with MRSs projected from f-structure.
- Output is generated by the English Resource Grammar (HPSG; Flickinger 2000)
- Further projects have built LOGON-based MT systems for other language pairs, notably JaEn (Bond et al 2011)

# Vauquois Pyramid (ObMT Triangle)

---

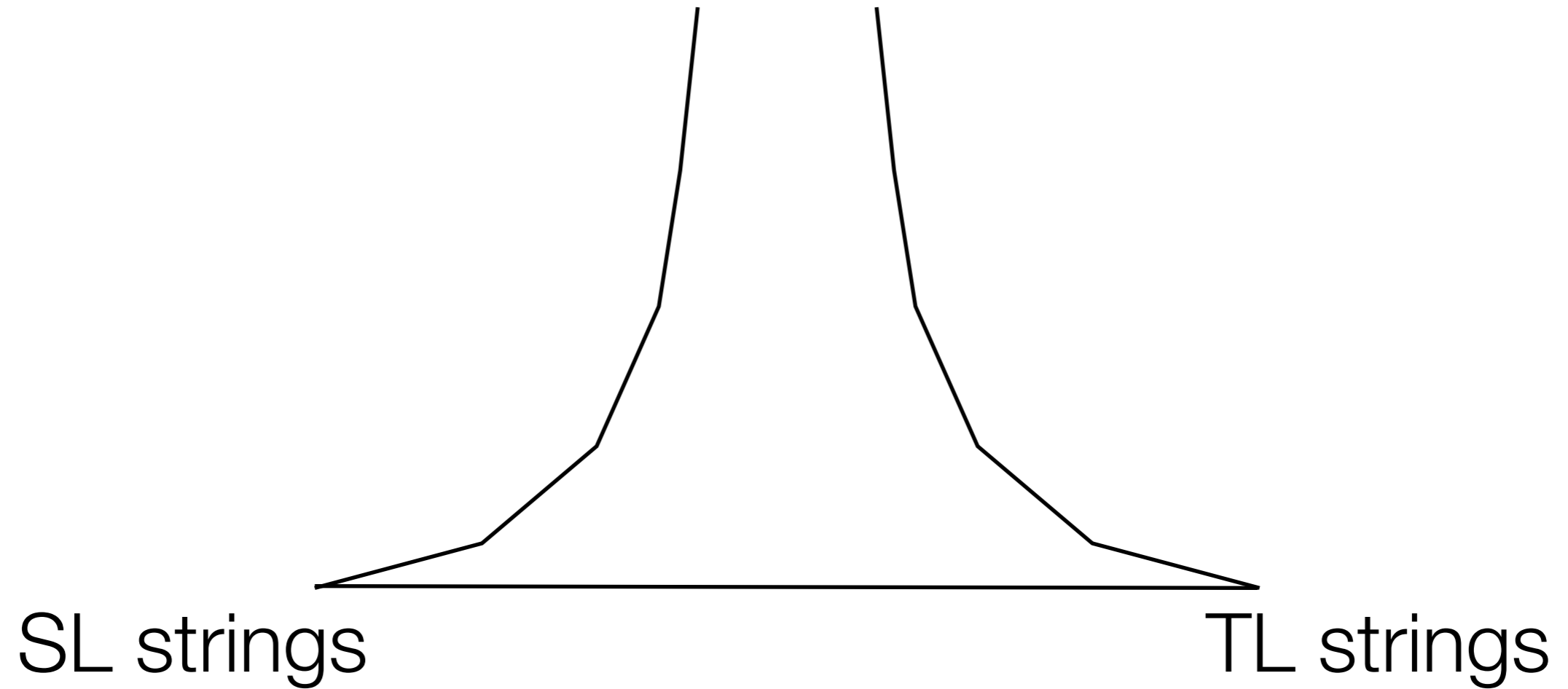


# Is MRS an interlingua?

- Could MRS be used to encode an interlingua?
- Could our grammars produce such an MRS-encoded interlingua?

# Copestake Volcano

---



# Mismatch: Translation divergences (Dorr 1994)

- **Categorial divergence:** Translation of words in one language into words that have a different part of speech in another language.
- **Conflational divergence:** The translation of two or more words in one language into one word in another language
- **Structural divergence:** The realization of verb arguments in different syntactic configurations in different languages.
- **Head swapping divergence:** The inversion of the structural dominance relation between two semantically equivalent words when translating from one language to another.
- **Thematic divergence:** The realization of verb arguments in different configurations that reflect different thematic to syntactic mapping orders.

# MRS 'harmonization' helps

- Just because it's not an interlingua doesn't mean the grammars can't be brought closer together.
- Example 1: Demonstratives (adjectives v. determiners)
- Example 2: COG-ST et al, reduction in quantifier-rel inventory
- Further potential for harmonization: pronouns v. pro-drop (but cf. information structure marking on overt pronouns)
- Other examples?

# LOGON processing steps

- Parse in source language
  - visualization tools for parses and MRSs
- Apply transfer grammar to produce new MRS
  - visualization tools for transfer outputs
- Generate in target language from new MRSs
  - visualization tools for input MRSs
  - compare to MRS produced by parsing expected output
  - generator chart

# Anatomy of a transfer rule

- Quadruple: [CONTEXT:] INPUT [!FILTER] -> OUTPUT
- Each item above is a (partial) MRS
- Rules apply to complete MRSs to produce partially rewritten MRSs.
- Resource sensitive: INPUT is consumed in producing OUTPUT.
- CONTEXT: Additional properties beyond the INPUT that must be satisfied. (Not consumed.)
- FILTER: Negative constraints; contexts in which the rule should not apply.

# Anatomy of a transfer rule

- Rules can be obligatory or optional.
- Optional rules produce non-determinism in the transfer process.
- Pairing each optional rule with one obligatory rule cuts down the transfer search space.
- Rules can also be grouped into sets for ‘extrinsic’ ordering (which we probably won’t need).
- Handled with chart-based processing.

## Example type

---

```
monotonic_mtr := mrs_transfer_rule &  
[ CONTEXT.HOOK.LTOP #h,  
  INPUT.HOOK.LTOP #h,  
  OUTPUT.HOOK.LTOP #h ].
```

# Example rule instance

---

```
pro-insert-arg1-mtr := monotonic_mtr &
[ INPUT.RELS <! !>,
  CONTEXT.RELS <! [ ARG0.SF prop-or-ques,
                    ARG1 #x & x ] !>,
  FILTER.RELS <! [ ARG0 #x ] !>,
  OUTPUT [ RELS <! [ PRED "_pronoun_n_rel",
                    ARG0 #x,
                    LBL #larg ],
          [ PRED "exist_q_rel",
            ARG0 #x,
            RSTR #harg ] !>,
          HCONS <! qeq &
                [ HARG #harg,
                  LARG #larg ] !> ],
  FLAGS.EQUAL < #x > ].
```

# Other transfer rules

---

- ==> Demo acm.tdl