DMRS Algebra

Guy Emerson

Towards a DMRS Algebra

- Two starting points:
  - Ann Copestake’s draft DMRS Algebra
  - Jonas Groschwitz et al.’s AM Algebra

Towards a DMRS Algebra

- Two starting points:
  - Ann Copestake’s draft DMRS Algebra
  - Jonas Groschwitz et al.’s AM Algebra
- Main aim of this talk:
  - DMRS Algebra expressed like AM Algebra

Motivations

- Why DMRS?
  - Constrain composition in Delph-in grammars

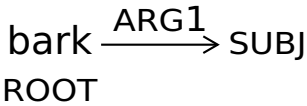
Motivations

- Why DMRS?
  - Constrain composition in Delph-in grammars (and maybe in neural parsers too)

Motivations

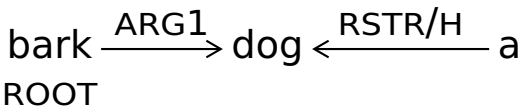
- Why DMRS?
  - Constrain composition in Delph-in grammars (and maybe in neural parsers too)
- Why AM?
  - Encourage cross-framework metalanguage
  - Borrow nice notation

Apply Operation



Apply Operation

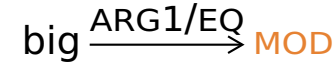
Apply Operation

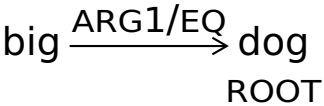


Modify Operation

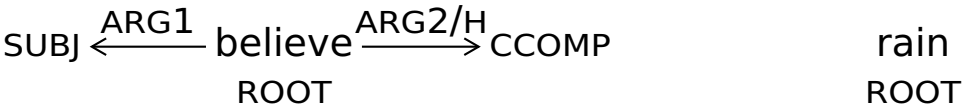
Modify Operation

Modify Operation





Apply Operation



Apply Operation

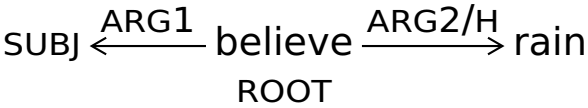
rain

ROOT

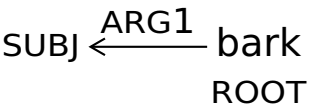
Apply Operation

rain

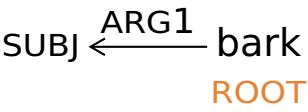
ROOT



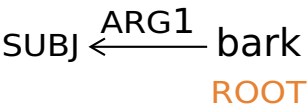
Apply Operation & Typing

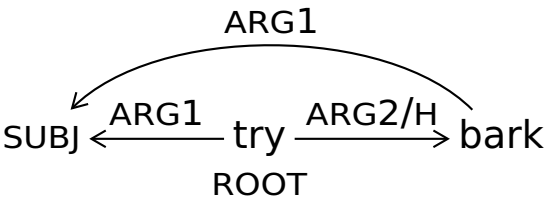


Apply Operation & Typing

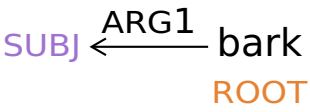


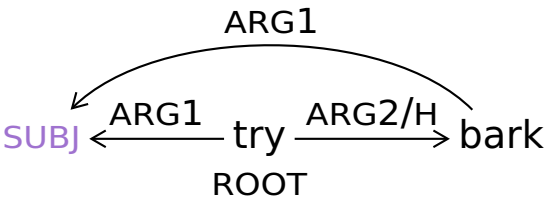
Apply Operation & Typing



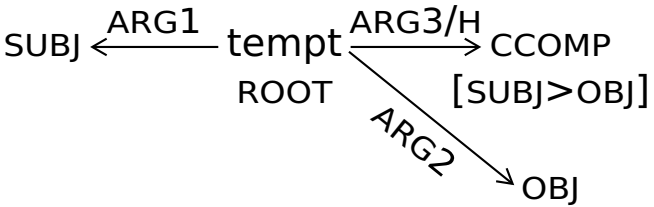


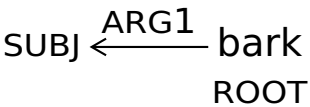
Apply Operation & Typing



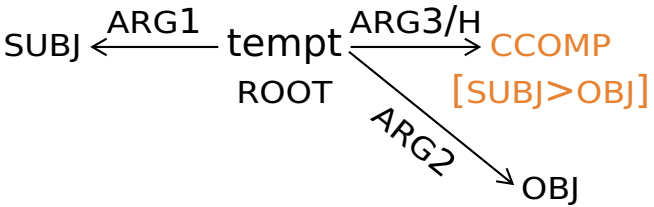


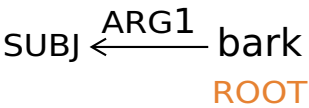
Apply Operation & Typing



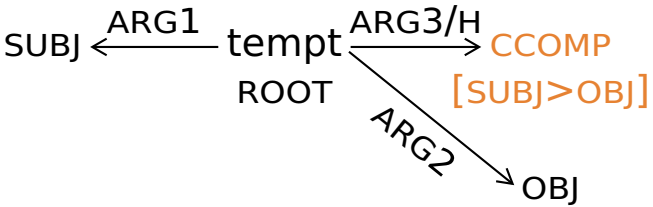


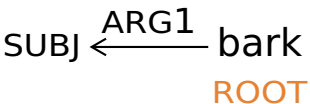
Apply Operation & Typing

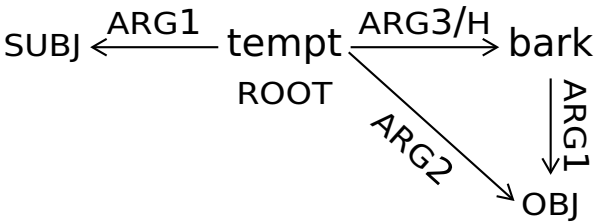




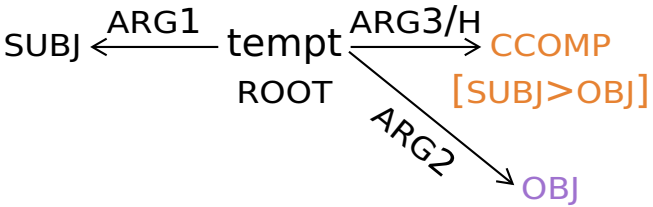
Apply Operation & Typing

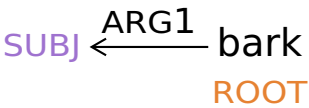


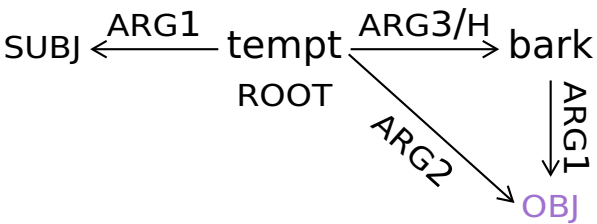




Apply Operation & Typing







AM Algebra Summary

- Every graph has a *root* and possibly some *slots*
- Two operations: *Apply* and *Modify*
- Functor graph controls slots in argument graph

AM and MRS Common Ground

- Slot-filling as a basic mechanism
- Typing corresponds to grammar principles

AM Algebra Not Enough For DMRS

- Lindemann, Groschwitz, Koller (2019):
  24% of EDS graphs in Open SDP 1.2
  not decomposable using AM Algebra

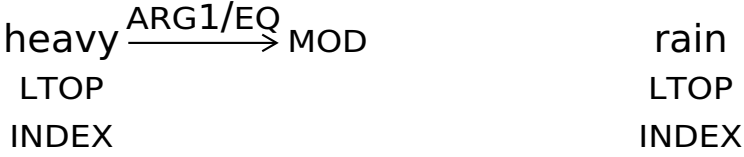
AM Algebra Not Enough For DMRS

- Lindemann, Groschwitz, Koller (2019):
  24% of EDS graphs in Open SDP 1.2
  not decomposable using AM Algebra
- DMRS adds scope to EDS,
  cannot be easier

What is Different in DMRS?

- Two “roots”: LTOP and INDEX
- Predicates from syntax (CCONT)
- Derivation tree given by grammar
- Special XARG slot
- Unary relabelling operations
- SLASH propagation
- Relaxed typing
- (Postprocessing of /NEQ)

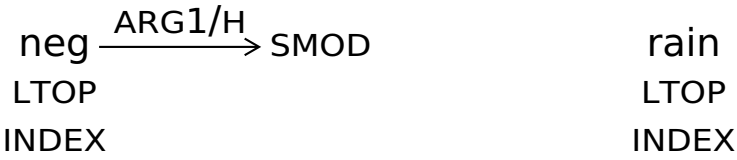
Nonscopal Modification



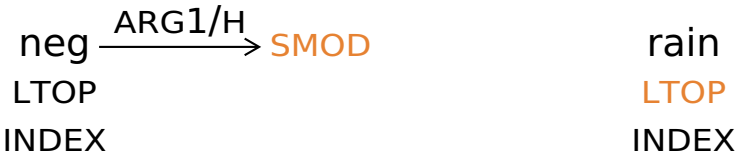
Nonscopal Modification

Nonscopal Modification

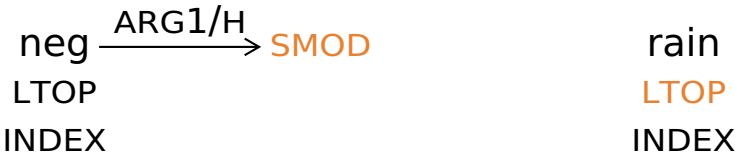
Scopal Modification



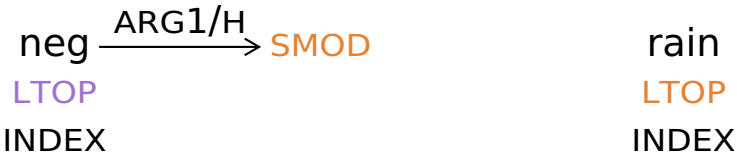
Scopal Modification

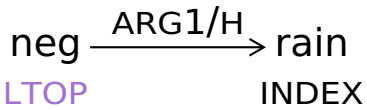


Scopal Modification

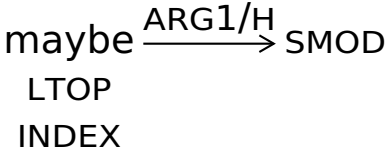


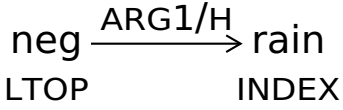
Scopal Modification



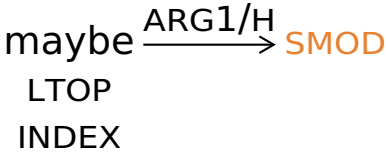


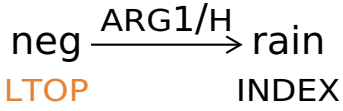
Scopal Modification



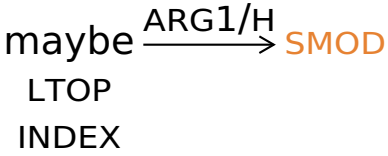


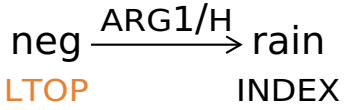
Scopal Modification

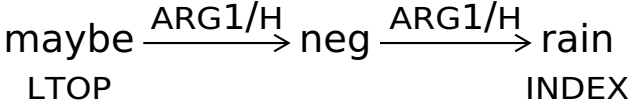




Scopal Modification







DMRS Composition Operations

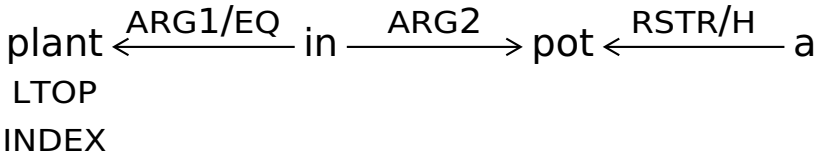
Operation	Slot filled by	INDEX from	LTOP from
Apply, nonscopal	INDEX	functor	functor
Apply, scopal	LTOP	functor	functor
Modify, nonscopal	INDEX	argument	argument
Modify, scopal	LTOP	argument	functor

DMRS Composition Operations

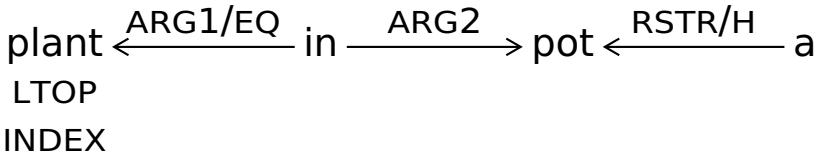
Operation	Slot filled by	INDEX from	LTOP from
Apply, nonscopal	INDEX	functor	functor
Apply, scopal	LTOP	functor	functor
Modify, nonscopal	INDEX	argument	argument
Modify, scopal	LTOP	argument	functor

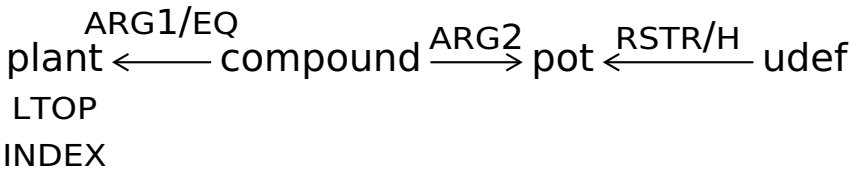
- Scopal arguments have /H (also written /QEQ)

CCONT: Compounds vs. Prepositions

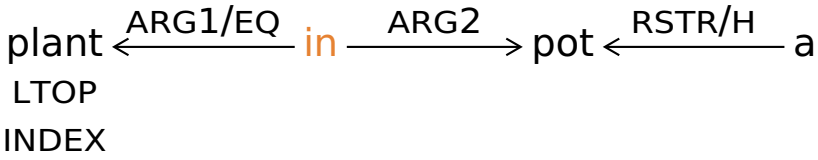


CCONT: Compounds vs. Prepositions

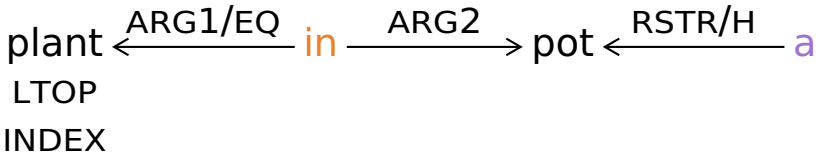




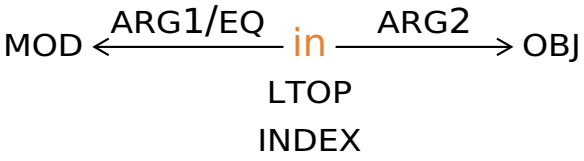
CCONT: Compounds vs. Prepositions



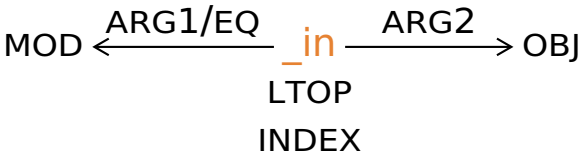
CCONT: Compounds vs. Prepositions



CCONT: Compounds vs. Prepositions



CCONT: Compounds vs. Prepositions



Derivation Trees

- AM Dependency Trees separate:
  - Composition (AM Algebra)
  - Derivation (AM Dependencies)

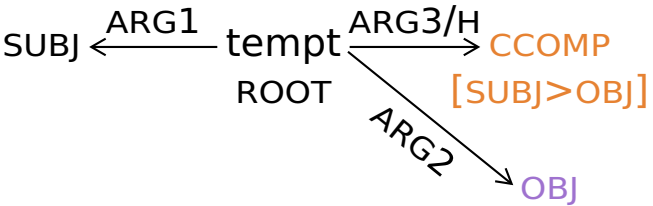
Derivation Trees

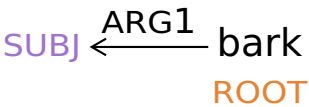
- AM Dependency Trees separate:
  - Composition (AM Algebra)
  - Derivation (AM Dependencies)
    [*not* semantic dependencies]

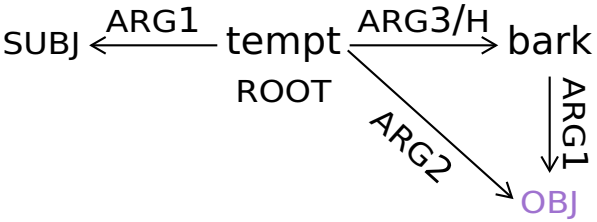
Derivation Trees

- AM Dependency Trees separate:
  - Composition (AM Algebra)
  - Derivation (AM Dependencies)
    [*not* semantic dependencies]
- Grammars provide derivation trees
  - … which may interfere with AM Algebra typing

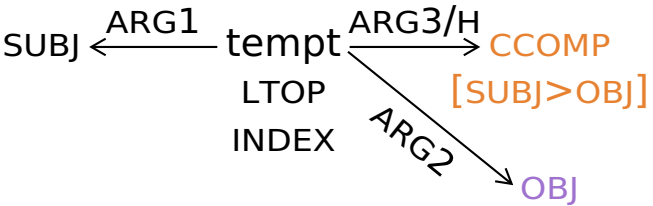
Fixed Derivations & Eager Composition

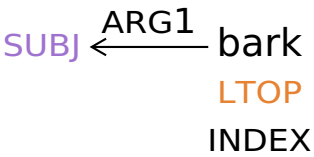


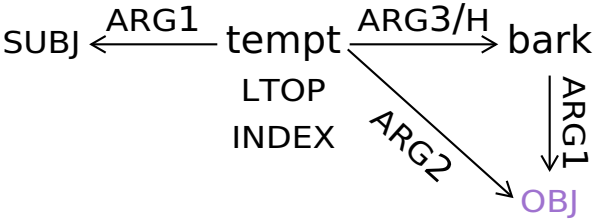




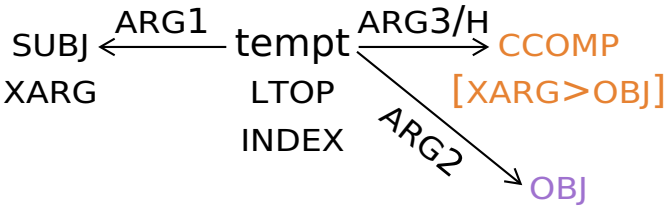
Fixed Derivations & Eager Composition

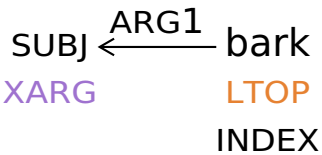


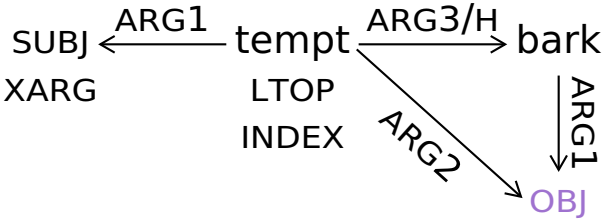




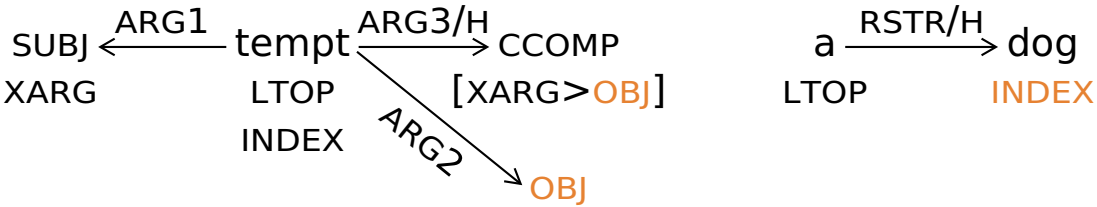
Fixed Derivations & Eager Composition



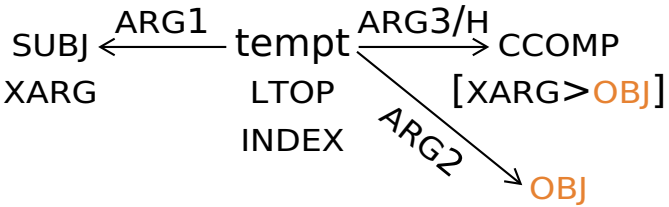


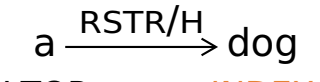


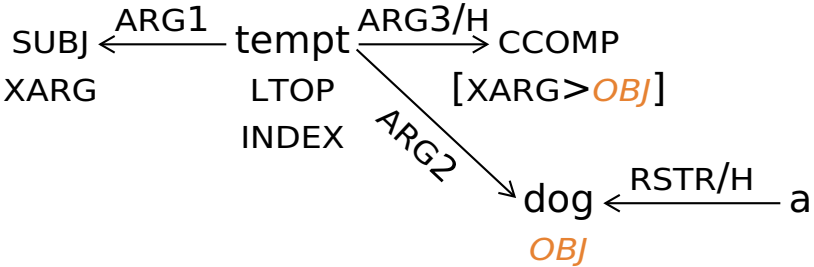
Fixed Derivations & Eager Composition



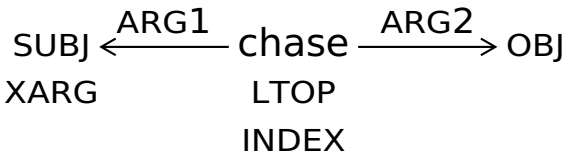
Fixed Derivations & Eager Composition



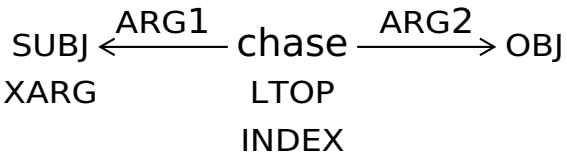


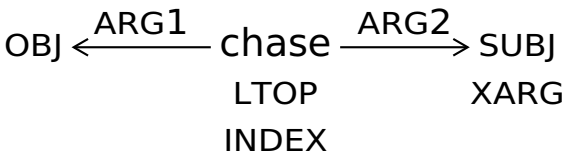


Relabelling Operations

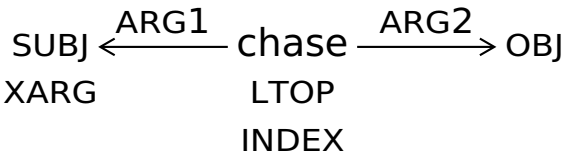


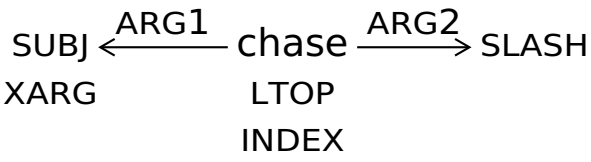
Relabelling Operations



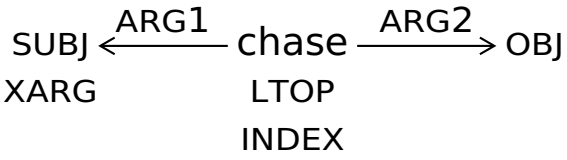


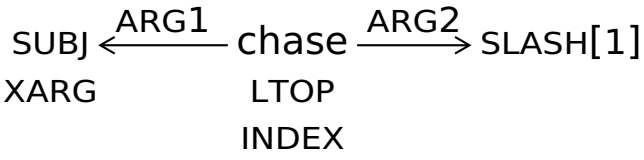
Relabelling Operations



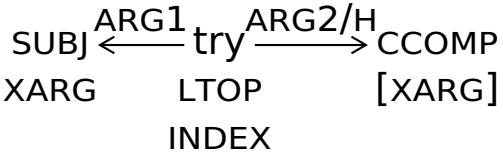


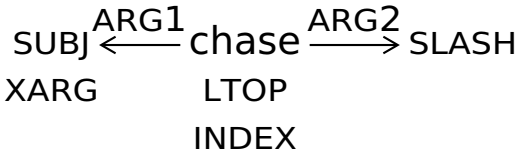
Relabelling Operations



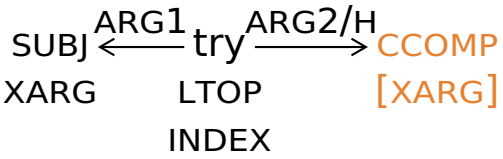


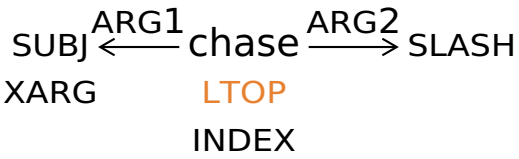
SLASH Propagation



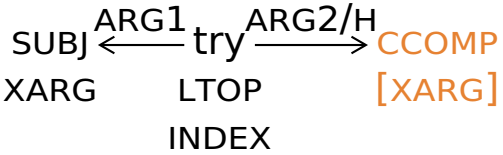


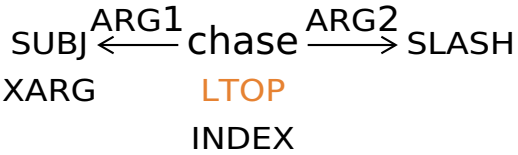
SLASH Propagation

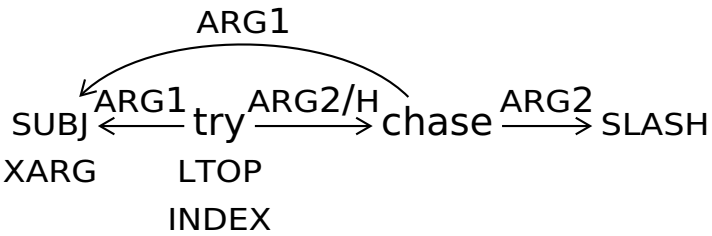




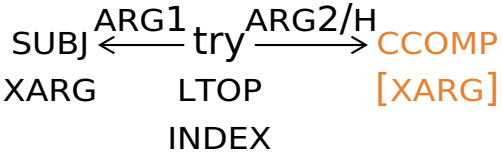
SLASH Propagation

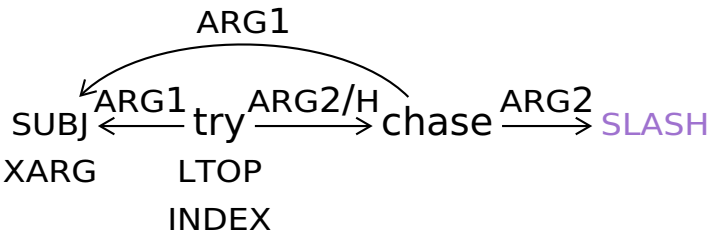




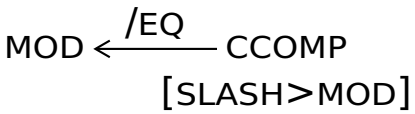


SLASH Propagation

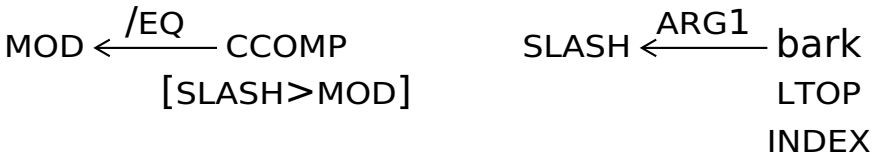




Relative Clauses

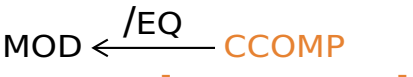


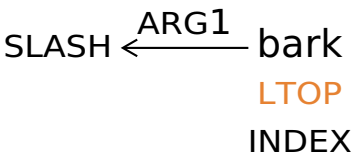
Relative Clauses

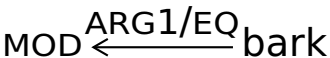


Relative Clauses

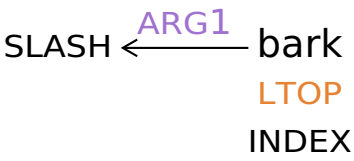
Relative Clauses





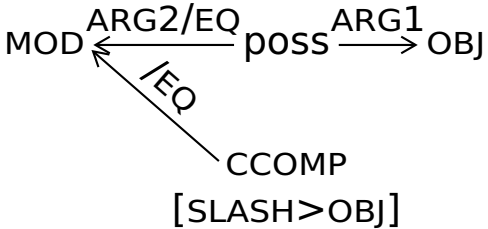


Relative Clauses

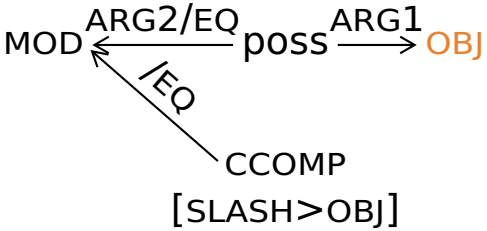


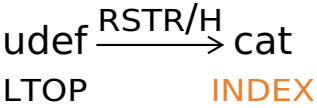
Relative Clauses

Possessive Relative Clauses

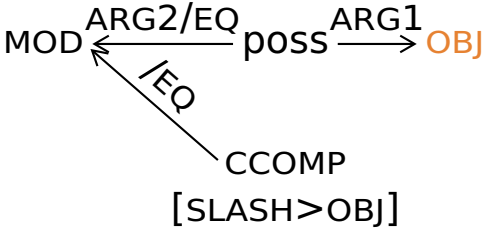


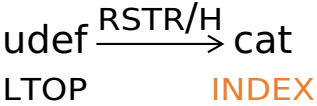
Possessive Relative Clauses

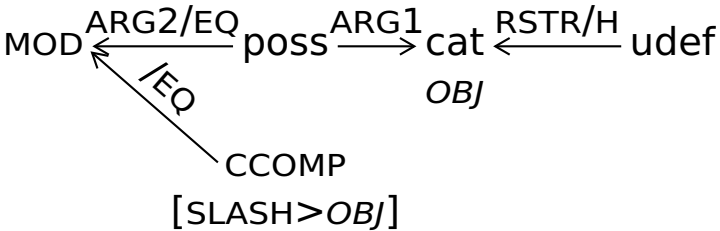




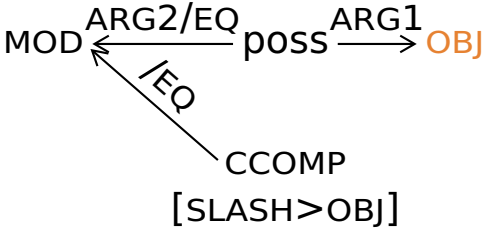
Possessive Relative Clauses

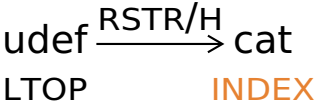


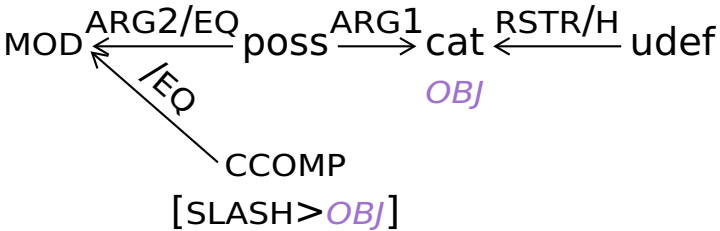




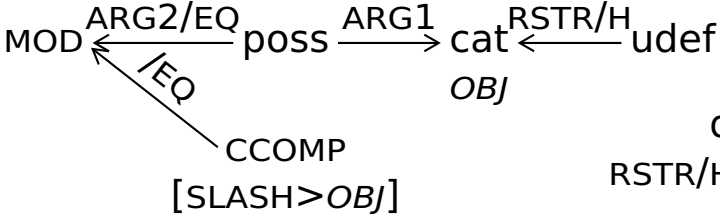
Possessive Relative Clauses

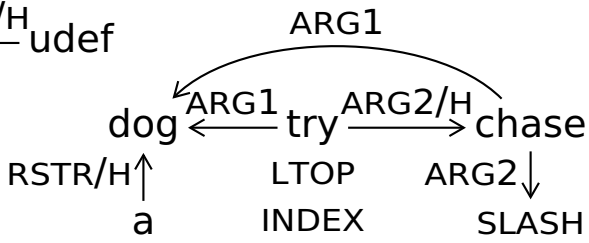




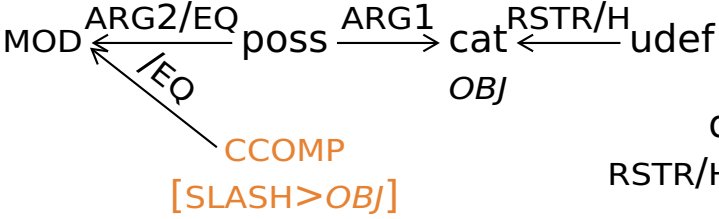


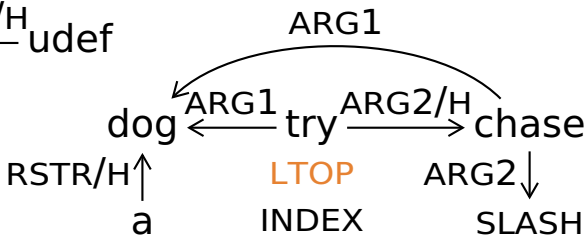
Possessive Relative Clauses



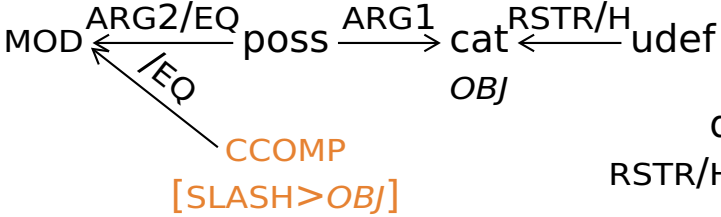


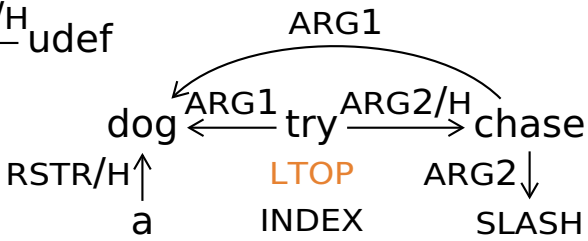
Possessive Relative Clauses

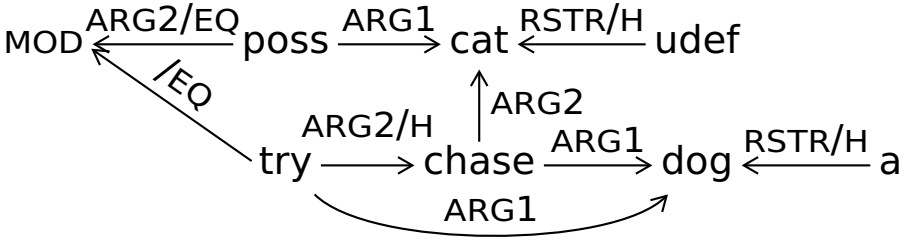




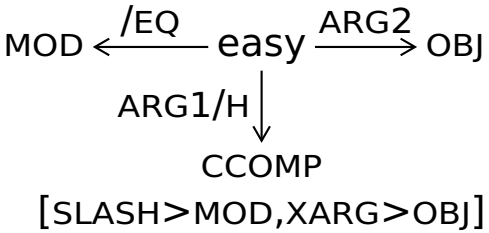
Possessive Relative Clauses

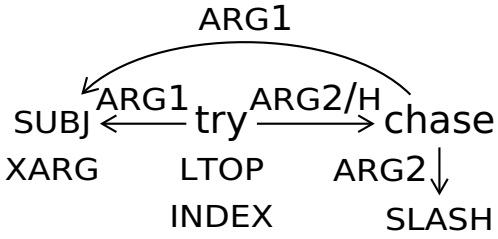




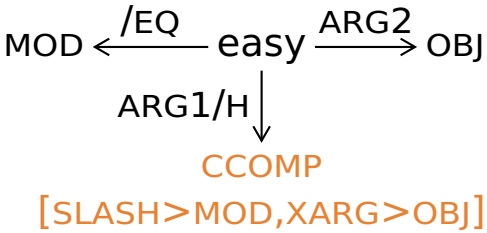


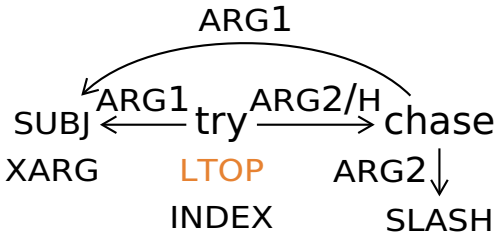
Easy Adjectives



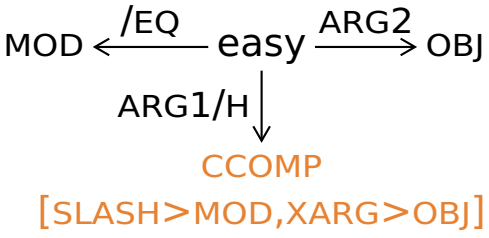


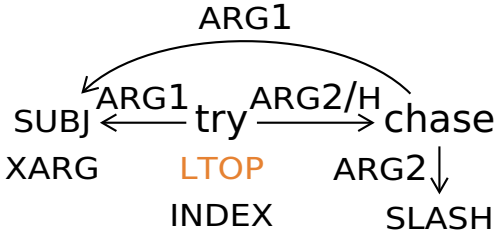
Easy Adjectives

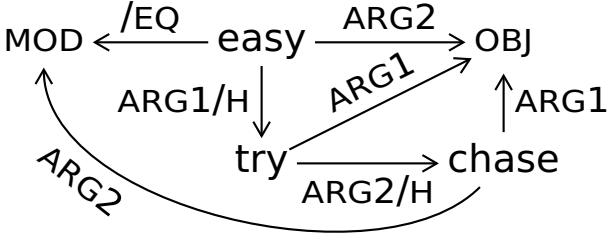




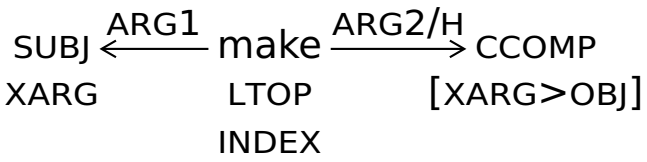
Easy Adjectives

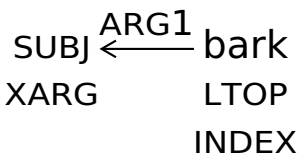




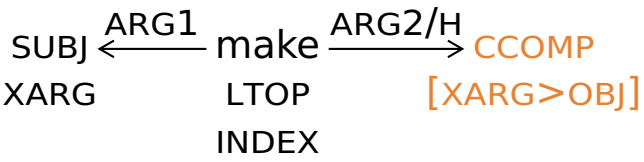


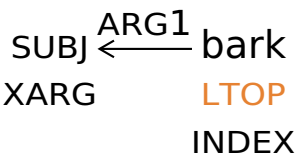
Relaxed Typing: Causatives



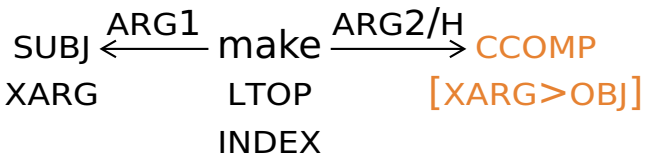


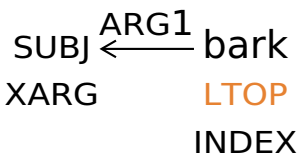
Relaxed Typing: Causatives

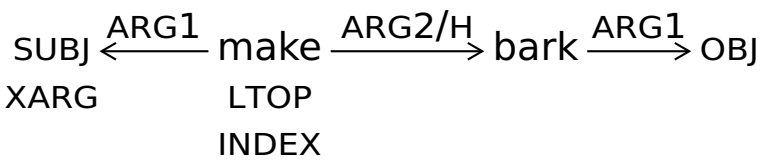




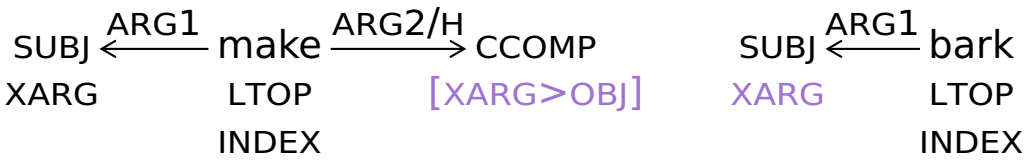
Relaxed Typing: Causatives



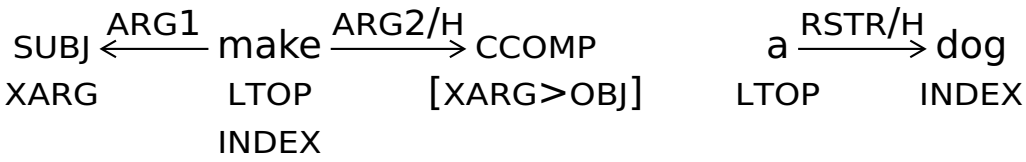




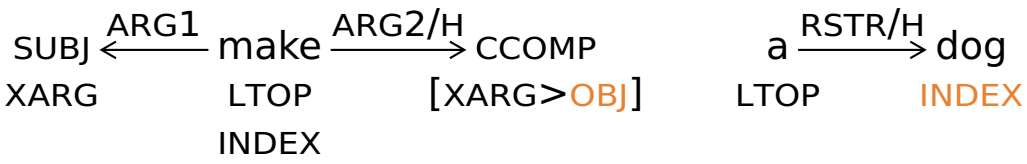
Relaxed Typing: Causatives



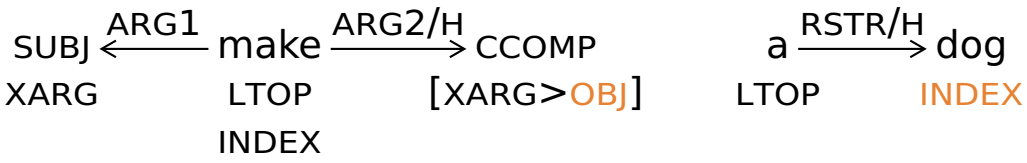
Relaxed Typing & Eager Composition

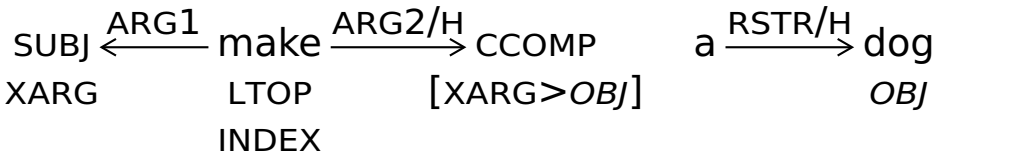


Relaxed Typing & Eager Composition

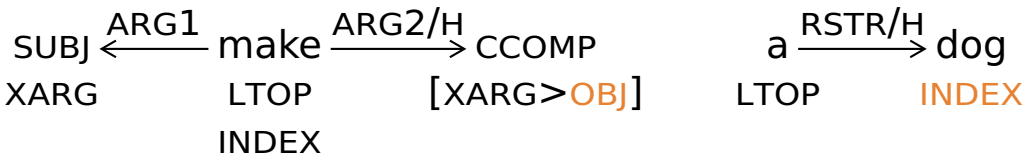


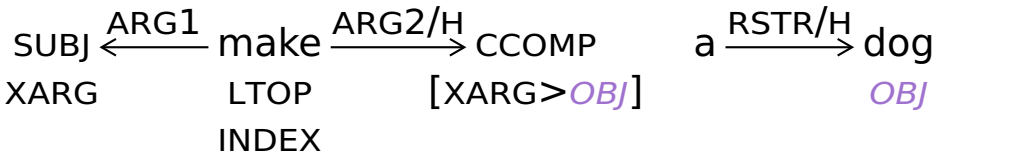
Relaxed Typing & Eager Composition



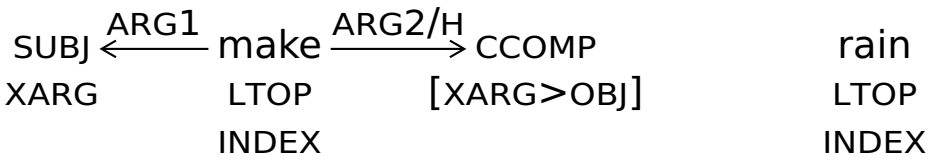


Relaxed Typing & Eager Composition

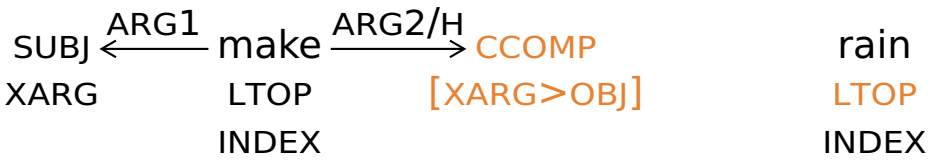




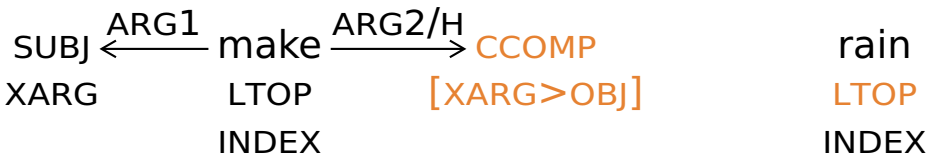
Relaxed Typing: Expletive Pronouns

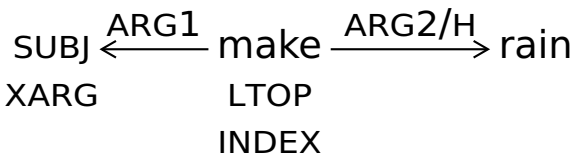


Relaxed Typing: Expletive Pronouns

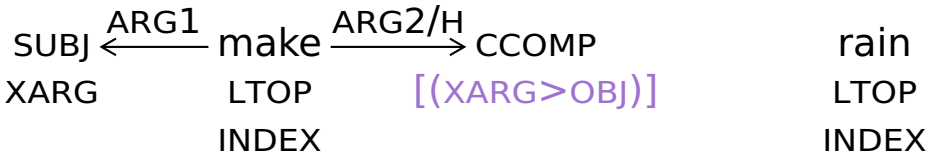


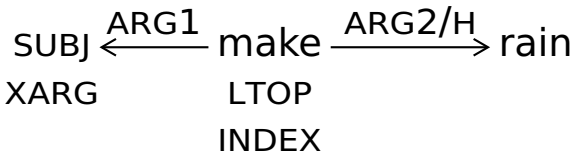
Relaxed Typing: Expletive Pronouns





Relaxed Typing: Expletive Pronouns





Summary

- AM Algebra helpful guide for formalising DMRS Algebra

Summary

- AM Algebra helpful guide for formalising DMRS Algebra
- Requires linguistically interesting extensions!