

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

bark $\xrightarrow{\text{ARG1}}$ SUBJ
ROOT

dog $\xleftarrow{\text{RSTR/H}}$ a
ROOT

Apply Operation

bark $\xrightarrow{\text{ARG1}}$ SUBJ
ROOT

dog $\xleftarrow{\text{RSTR/H}}$ a
ROOT

Apply Operation

bark $\xrightarrow{\text{ARG1}}$ SUBJ
ROOT

dog $\xleftarrow{\text{RSTR/H}}$ a
ROOT

bark $\xrightarrow{\text{ARG1}}$ dog $\xleftarrow{\text{RSTR/H}}$ a
ROOT

Modify Operation

big $\xrightarrow{\text{ARG1/EQ}}$ MOD
ROOT

dog
ROOT

Modify Operation

big $\xrightarrow{\text{ARG1/EQ}}$ MOD
ROOT

dog
ROOT

Modify Operation

big $\xrightarrow{\text{ARG1/EQ}}$ MOD
ROOT

dog
ROOT

big $\xrightarrow{\text{ARG1/EQ}}$ dog
ROOT

Apply Operation

SUBJ $\xleftarrow{\text{ARG1}}$ believe $\xrightarrow{\text{ARG2/H}}$ CCOMP
ROOT

rain
ROOT

Apply Operation

SUBJ $\xleftarrow{\text{ARG1}}$ believe $\xrightarrow{\text{ARG2/H}}$ CCOMP
ROOT

rain
ROOT

Apply Operation

SUBJ $\xleftarrow{\text{ARG1}}$ believe $\xrightarrow{\text{ARG2/H}}$ CCOMP
ROOT

rain
ROOT

SUBJ $\xleftarrow{\text{ARG1}}$ believe $\xrightarrow{\text{ARG2/H}}$ rain
ROOT

Apply Operation & Typing

SUBJ ←^{ARG1} try ^{ARG2/H}→ CCOMP
 ROOT [SUBJ]

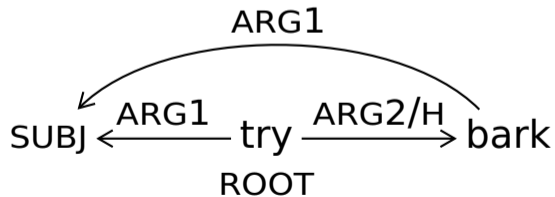
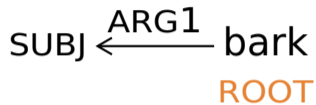
SUBJ ←^{ARG1} bark
 ROOT

Apply Operation & Typing

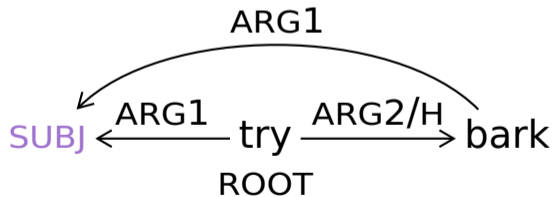
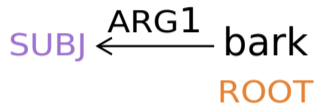
SUBJ ←^{ARG1} try ^{ARG2/H} → CCOMP
ROOT [SUBJ]

SUBJ ←^{ARG1} bark
ROOT

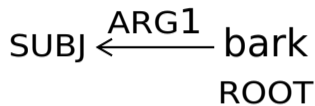
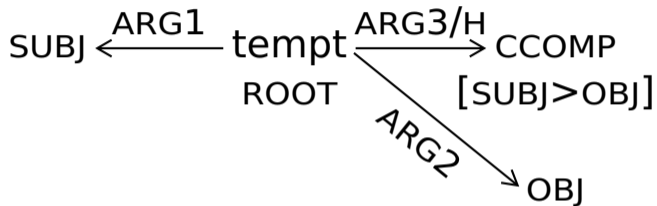
Apply Operation & Typing



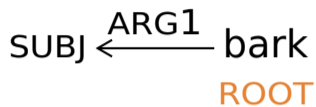
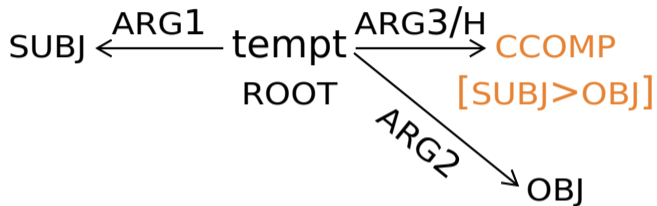
Apply Operation & Typing



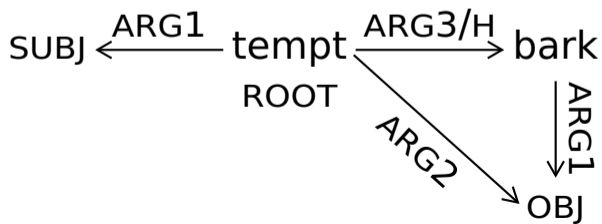
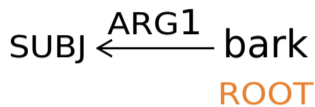
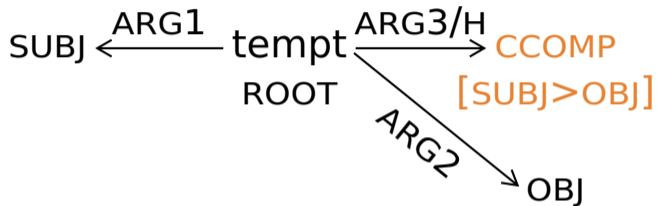
Apply Operation & Typing



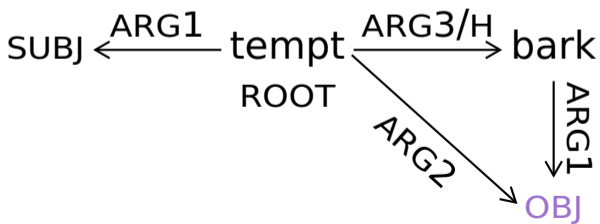
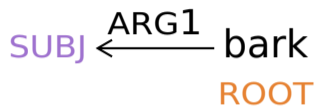
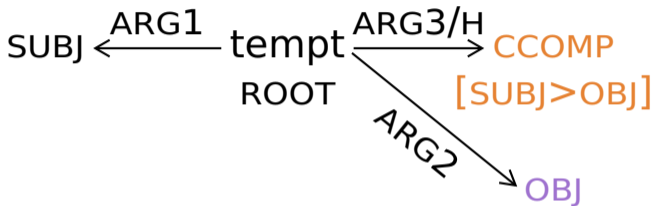
Apply Operation & Typing



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

heavy $\xrightarrow{\text{ARG1/EQ}}$ MOD
LTOP
INDEX

rain
LTOP
INDEX

Nonscopal Modification

heavy $\xrightarrow{\text{ARG1/EQ}}$ MOD
LTOP
INDEX

rain
LTOP
INDEX

Nonscopal Modification

heavy $\xrightarrow{\text{ARG1/EQ}}$ MOD
LTOP
INDEX

rain
LTOP
INDEX

heavy $\xrightarrow{\text{ARG1/EQ}}$ rain
LTOP
INDEX

Scopal Modification

neg $\xrightarrow{\text{ARG1/H}}$ SMOD
LTOP
INDEX

rain
LTOP
INDEX

Scopal Modification

neg $\xrightarrow{\text{ARG1/H}}$ SMOD
LTOP
INDEX

rain
LTOP
INDEX

Scopal Modification

neg $\xrightarrow{\text{ARG1/H}}$ SMOD
LTOP
INDEX

rain
LTOP
INDEX

neg $\xrightarrow{\text{ARG1/H}}$ rain
LTOP INDEX

Scopal Modification

neg $\xrightarrow{\text{ARG1/H}}$ SMOD
LTOP
INDEX

rain
LTOP
INDEX

neg $\xrightarrow{\text{ARG1/H}}$ rain
LTOP INDEX

Scopal Modification

maybe $\xrightarrow{\text{ARG1/H}}$ SMOD
LTOP
INDEX

neg $\xrightarrow{\text{ARG1/H}}$ rain
LTOP INDEX

Scopal Modification

maybe $\xrightarrow{\text{ARG1/H}}$ SMOD
LTOP
INDEX

neg $\xrightarrow{\text{ARG1/H}}$ rain
LTOP INDEX

Scopal Modification

maybe $\xrightarrow{\text{ARG1/H}}$ SMOD
LTOP
INDEX

neg $\xrightarrow{\text{ARG1/H}}$ rain
LTOP INDEX

maybe $\xrightarrow{\text{ARG1/H}}$ neg $\xrightarrow{\text{ARG1/H}}$ rain
LTOP INDEX

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

plant ←^{ARG1/EQ} in —^{ARG2}→ pot ←^{RSTR/H} a
LTOP
INDEX

CCONT: Compounds vs. Prepositions

plant $\xleftarrow{\text{ARG1/EQ}}$ in $\xrightarrow{\text{ARG2}}$ pot $\xleftarrow{\text{RSTR/H}}$ a
LTOP
INDEX

plant $\xleftarrow{\text{ARG1/EQ}}$ compound $\xrightarrow{\text{ARG2}}$ pot $\xleftarrow{\text{RSTR/H}}$ udef
LTOP
INDEX

CCONT: Compounds vs. Prepositions

plant ←^{ARG1/EQ} in →^{ARG2} pot ←^{RSTR/H} a
LTOP
INDEX

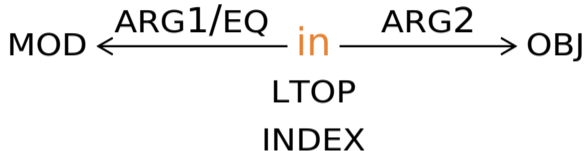
plant ←^{ARG1/EQ} compound →^{ARG2} pot ←^{RSTR/H} udef
LTOP
INDEX

CCONT: Compounds vs. Prepositions

plant ←^{ARG1/EQ} in →^{ARG2} pot ←^{RSTR/H} a
LTOP
INDEX

plant ←^{ARG1/EQ} compound →^{ARG2} pot ←^{RSTR/H} udef
LTOP
INDEX

CCONT: Compounds vs. Prepositions



CCONT: Compounds vs. Prepositions

MOD ← ARG1/EQ _in ARG2 → OBJ
LTOP
INDEX

MOD ← ARG1/EQ compound ARG2 → OBJ
LTOP
INDEX

Derivation Trees

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

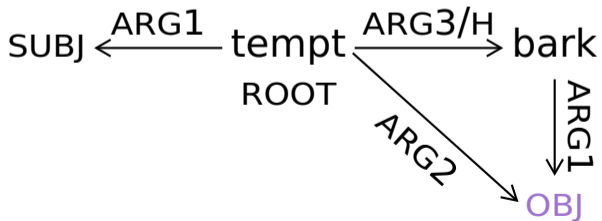
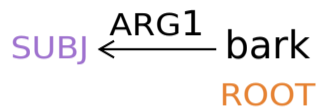
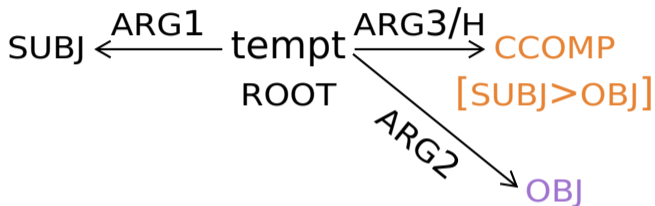
Derivation Trees

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

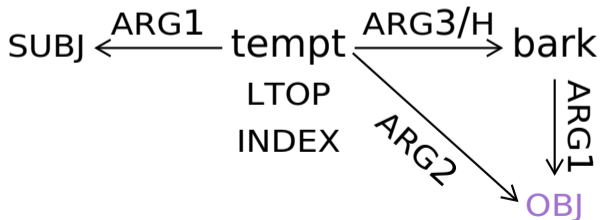
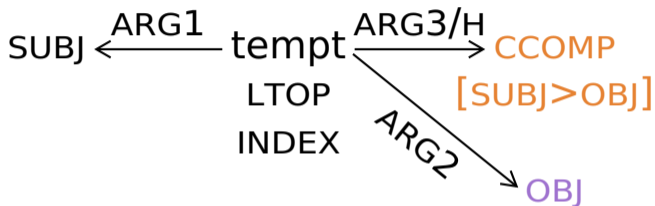
Derivation Trees

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

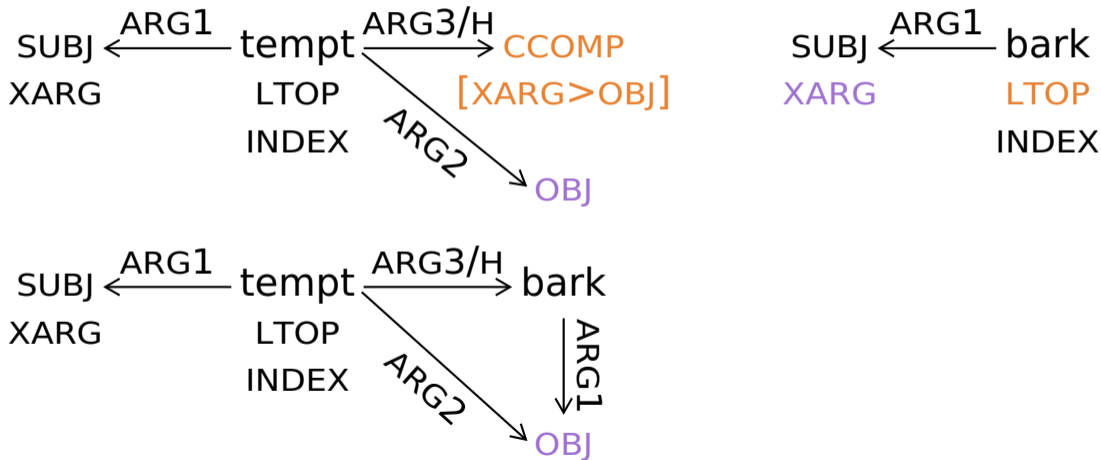
Fixed Derivations & Eager Composition



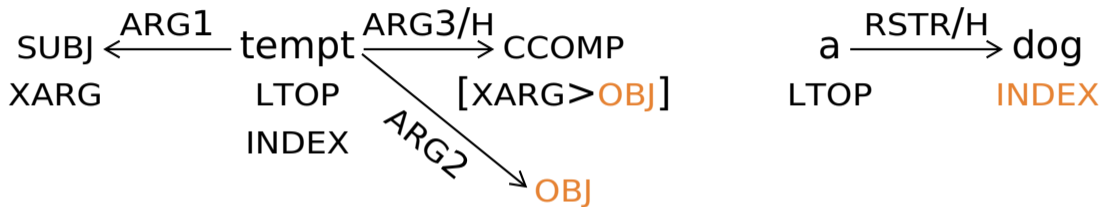
Fixed Derivations & Eager Composition



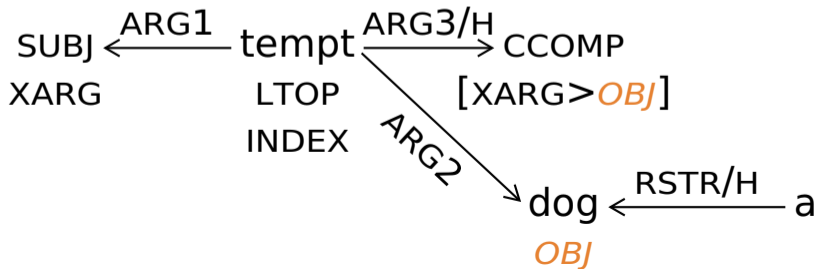
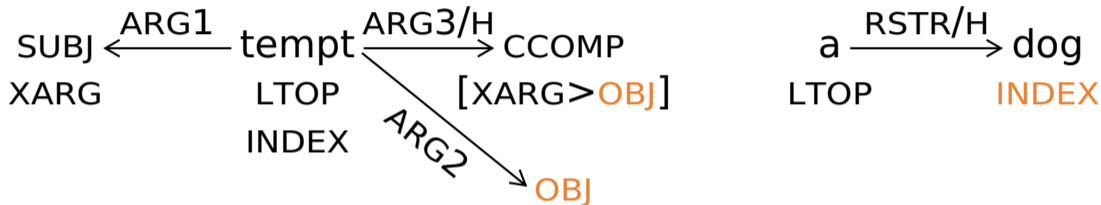
Fixed Derivations & Eager Composition



Fixed Derivations & Eager Composition



Fixed Derivations & Eager Composition



Relabelling Operations



Relabelling Operations

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ OBJ
XARG LTOP
 INDEX

OBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ SUBJ
 LTOP XARG
 INDEX

Relabelling Operations

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ OBJ
XARG LTOP
 INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH
XARG LTOP
 INDEX

Relabelling Operations

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ OBJ
XARG LTOP
 INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH[1]
XARG LTOP
 INDEX

SLASH Propagation

SUBJ $\xleftarrow{\text{ARG1}}$ try $\xrightarrow{\text{ARG2/H}}$ CCOMP
XARG LTOP [XARG]
INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH
XARG LTOP
INDEX

SLASH Propagation

SUBJ $\xleftarrow{\text{ARG1}}$ try $\xrightarrow{\text{ARG2/H}}$ CCOMP
XARG LTOP [XARG]
INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH
XARG LTOP
INDEX

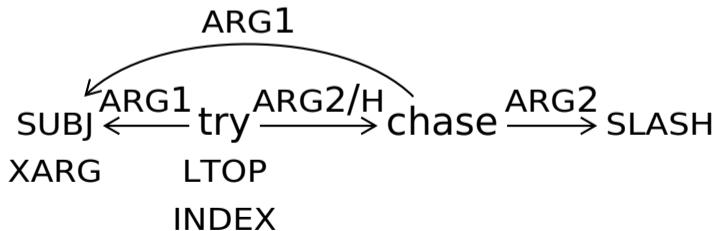
SLASH Propagation

SUBJ $\xleftarrow{\text{ARG1}}$ try $\xrightarrow{\text{ARG2/H}}$ CCOMP
XARG LTOP [XARG]
INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH
XARG LTOP
INDEX

ARG1

SUBJ $\xleftarrow{\text{ARG1}}$ try $\xrightarrow{\text{ARG2/H}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH
XARG LTOP
INDEX

A diagram illustrating the propagation of the SLASH property. It shows two phrases: 'try' and 'chase'. The 'try' phrase has a subject (SUBJ) and a left-top argument (LTOP) labeled 'try'. The 'chase' phrase has a left-top argument (LTOP) labeled 'chase' and a slash (SLASH). Both phrases share a common argument, ARG1, which is represented by a curved arrow pointing from the 'chase' phrase back to the 'try' phrase. The 'try' phrase also has a right-hand argument (ARG2/H) and a slash (SLASH). The 'chase' phrase has a right-hand argument (ARG2) and a slash (SLASH). The diagram shows how the SLASH property is propagated from the 'chase' phrase to the 'try' phrase through the shared ARG1 argument.

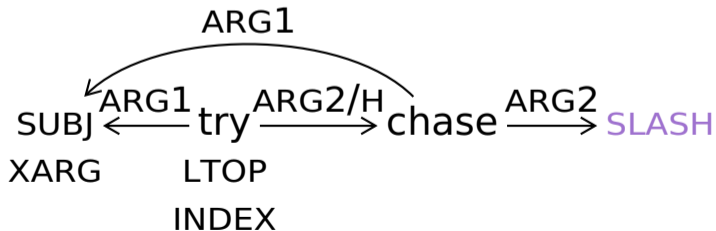
SLASH Propagation

SUBJ $\xleftarrow{\text{ARG1}}$ try $\xrightarrow{\text{ARG2/H}}$ CCOMP
XARG LTOP [XARG]
INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH
XARG LTOP
INDEX

ARG1

SUBJ $\xleftarrow{\text{ARG1}}$ try $\xrightarrow{\text{ARG2/H}}$ chase $\xrightarrow{\text{ARG2}}$ SLASH
XARG LTOP
INDEX

A diagram illustrating the propagation of the SLASH relation. It shows two phrases: 'try' and 'chase'. The 'try' phrase has a subject (SUBJ) and a left-top argument (LTOP) labeled 'try'. The 'chase' phrase has a left-top argument (LTOP) labeled 'chase' and a slash (SLASH). An arrow labeled 'ARG1' points from the 'try' phrase to the 'chase' phrase, indicating that the slash is propagated to the 'try' phrase through the shared ARG1 argument.

Relative Clauses

MOD ← ^{/EQ} — CCOMP
[SLASH>MOD]

Relative Clauses

MOD ←^{/EQ} CCOMP
[SLASH>MOD]

SLASH ←^{ARG1} bark
LTOP
INDEX

Relative Clauses

MOD ← ^{/EQ} CCOMP
[SLASH > MOD]

SLASH ← ^{ARG1} bark
LTOP
INDEX

Relative Clauses

MOD $\xleftarrow{/EQ}$ CCOMP
[SLASH > MOD]

SLASH $\xleftarrow{ARG1}$ bark
LTOP
INDEX

MOD $\xleftarrow{ARG1/EQ}$ bark

Relative Clauses

MOD ← ^{/EQ} CCOMP
[SLASH > MOD]

SLASH ← ^{ARG1} bark
LTOP
INDEX

MOD ← ^{ARG1/EQ} bark

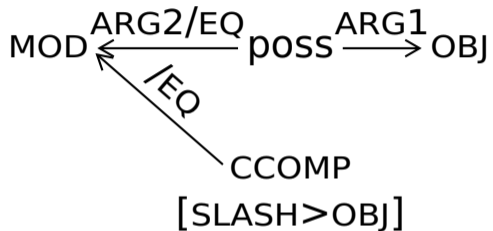
Relative Clauses

MOD ← ^{/EQ} CCOMP
XARG [SLASH>MOD]
[LTOP]
[INDEX]

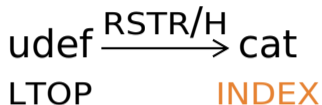
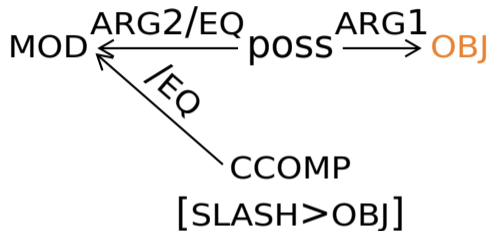
SLASH ← ^{ARG1} bark
LTOP
INDEX

MOD ← ^{ARG1/EQ} bark
XARG LTOP
INDEX

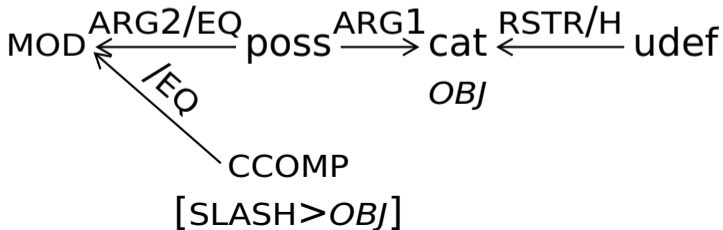
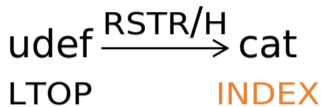
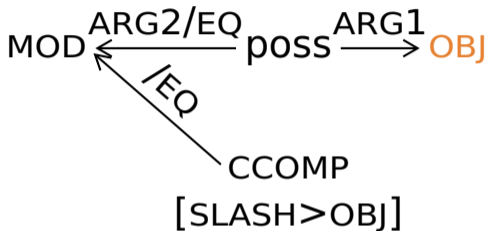
Possessive Relative Clauses



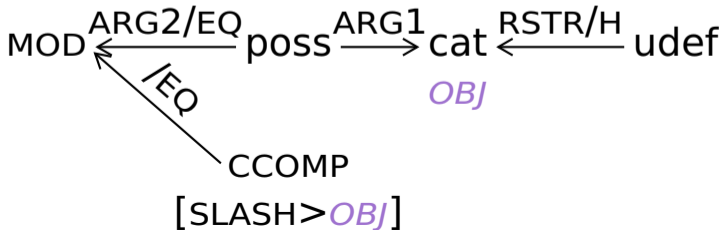
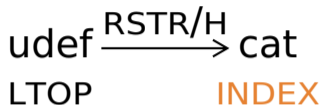
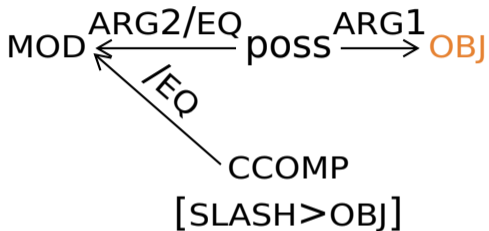
Possessive Relative Clauses



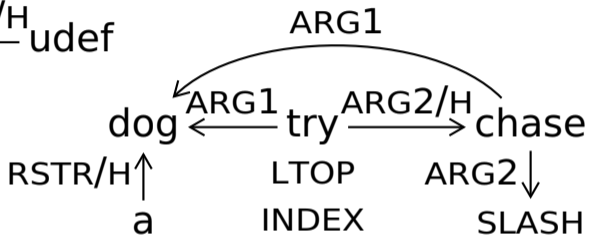
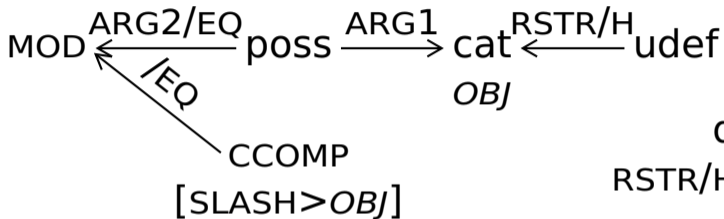
Possessive Relative Clauses



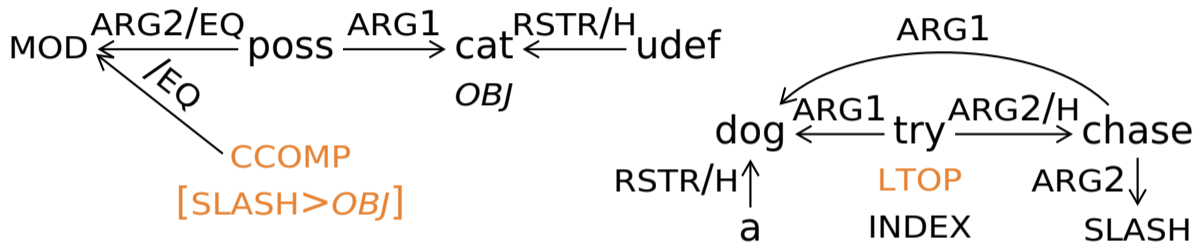
Possessive Relative Clauses



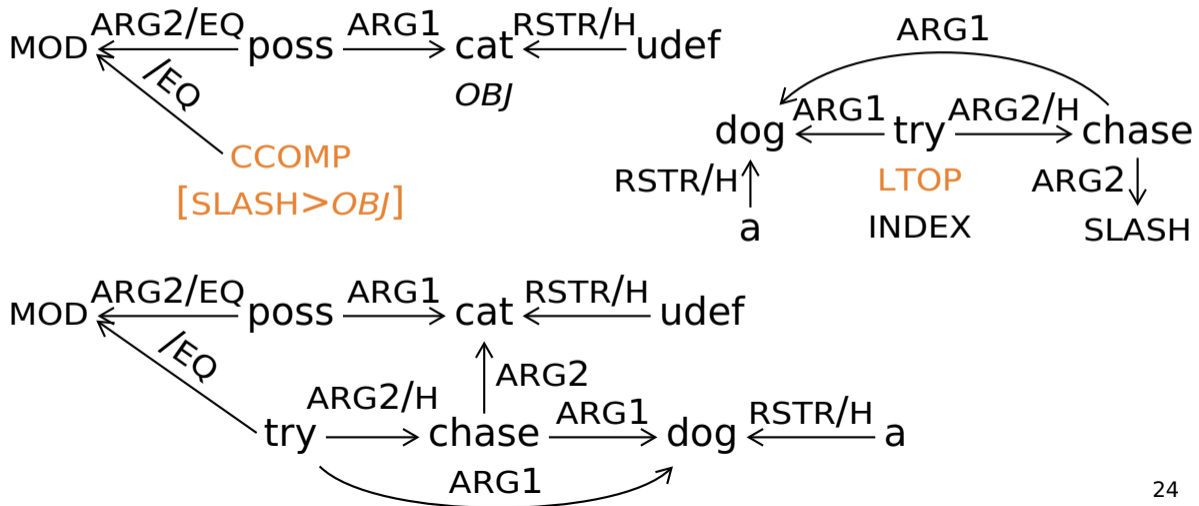
Possessive Relative Clauses



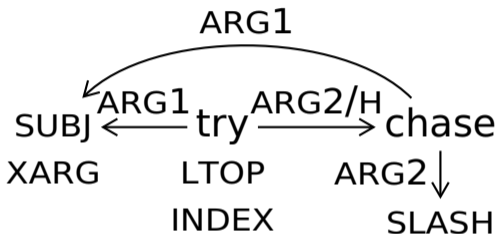
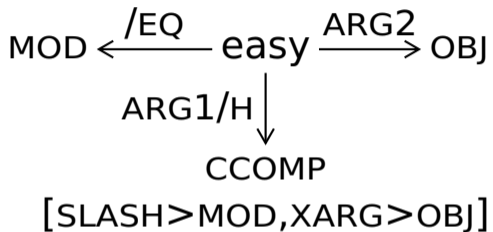
Possessive Relative Clauses



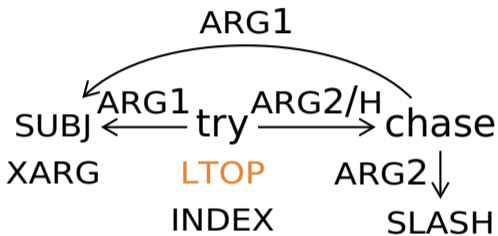
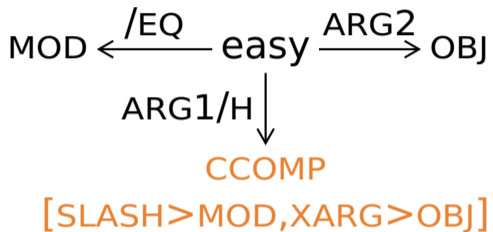
Possessive Relative Clauses



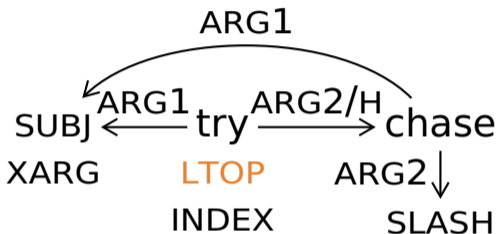
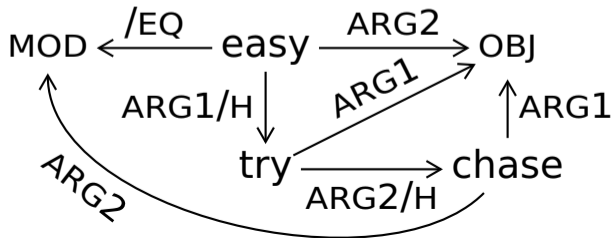
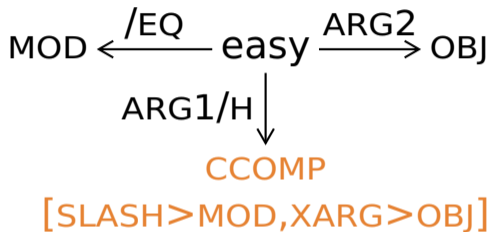
Easy Adjectives



Easy Adjectives



Easy Adjectives

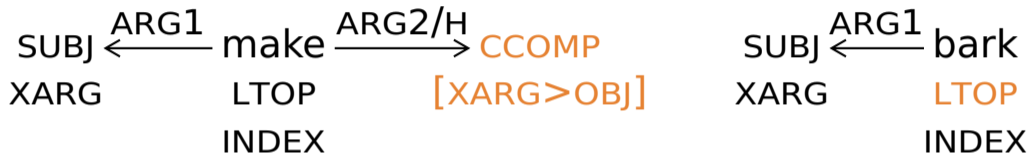


Relaxed Typing: Causatives

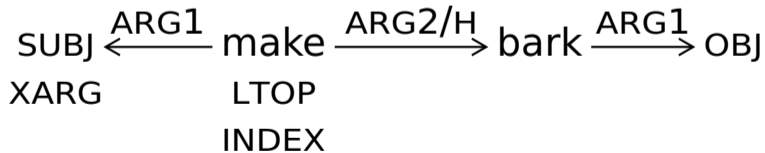
SUBJ $\xleftarrow{\text{ARG1}}$ make $\xrightarrow{\text{ARG2/H}}$ CCOMP
XARG LTOP [XARG>OBJ]
INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ bark
XARG LTOP
INDEX

Relaxed Typing: Causatives



Relaxed Typing: Causatives

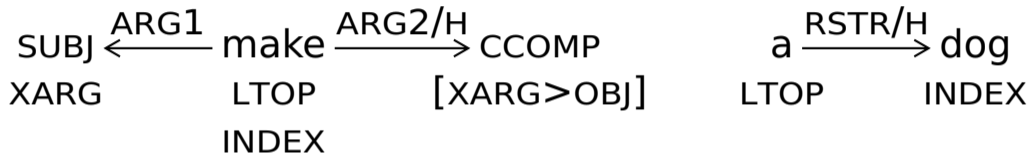


Relaxed Typing: Causatives

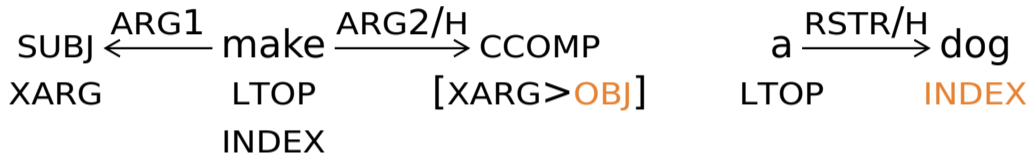
SUBJ	$\xleftarrow{\text{ARG1}}$	make	$\xrightarrow{\text{ARG2/H}}$	CCOMP	SUBJ	$\xleftarrow{\text{ARG1}}$	bark
XARG		LTOP		[XARG>OBJ]	XARG		LTOP
		INDEX					INDEX

SUBJ	$\xleftarrow{\text{ARG1}}$	make	$\xrightarrow{\text{ARG2/H}}$	bark	$\xrightarrow{\text{ARG1}}$	OBJ
XARG		LTOP				
		INDEX				

Relaxed Typing & Eager Composition



Relaxed Typing & Eager Composition



Relaxed Typing & Eager Composition

SUBJ $\xleftarrow{\text{ARG1}}$ make $\xrightarrow{\text{ARG2/H}}$ CCOMP a $\xrightarrow{\text{RSTR/H}}$ dog
XARG LTOP [XARG > OBJ] LTOP INDEX
INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ make $\xrightarrow{\text{ARG2/H}}$ CCOMP a $\xrightarrow{\text{RSTR/H}}$ dog
XARG LTOP [XARG > OBJ] OBJ
INDEX

Relaxed Typing: Expletive Pronouns



Relaxed Typing: Expletive Pronouns

SUBJ $\xleftarrow{\text{ARG1}}$ make $\xrightarrow{\text{ARG2/H}}$ CCOMP rain
XARG LTOP [XARG>OBJ] LTOP
INDEX INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ make $\xrightarrow{\text{ARG2/H}}$ rain
XARG LTOP
INDEX

Relaxed Typing: Expletive Pronouns

SUBJ $\xleftarrow{\text{ARG1}}$ make $\xrightarrow{\text{ARG2/H}}$ CCOMP rain
XARG LTOP [(XARG>OBJ)] LTOP
 INDEX INDEX

SUBJ $\xleftarrow{\text{ARG1}}$ make $\xrightarrow{\text{ARG2/H}}$ rain
XARG LTOP
 INDEX

Summary

- AM Algebra helpful guide for formalising DMRS Algebra

Summary

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