Barker—Szabolcsi, ESSLLI 2007

Days 4 and 5

Partially Ordered Categories: Optionality, Scope, and Licensing

based on Bernardi & Szabolcsi 2007

Part 1: Basic ideas

August 14

- It is beneficial to use partially ordered (sub)categories, as opposed to atomic ones
- The relevant ordering relation can be given by the derivability relation in some logic
- Applications to the general problems of optionality, scope, licensing, and to the specific problems of quantifier scope in Hungarian

- Thanks especially to Øystein Nilsen, Lucas Champollion, and grad students at NYU for discussions.

Three problems, one solution:

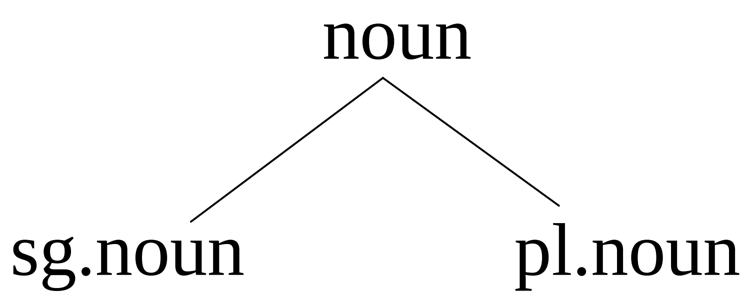
- Fine-grained subcategories

The dog/*dogs is hungry.

The *dog/dogs are hungry.

The dog/dogs will be hungry.

How can will be be indifferent to the number of the subject?



will be takes plain "nouns" as subject

sg.noun $\subseteq$ noun

pl.noun $\subseteq$ noun

- Optionality of category changing functors

I think Mary left

I think that Mary left

*I think that that Mary left

that $\notin$ $Cat_{X/X}$

How can think be indifferent to whether that is there?

CP

|

IP

think takes CP

IP $\subseteq$ CP

Mary left $\in$ IP

that Mary left $\in$ CP

- Licensing ungrammatical categories (Bernardi 2002)

Mary drank wine

*Mary drank any more wine

Whether Mary drank wine (is the question)

Whether Mary drank any more wine (is…)

How does whether rescue *Mary drank any …?

Complete S   Incomplete S

Good-enough S

whether takes Incompl. S

& returns Complete S

M drank any more wine $\in$ Incompl.S

M drank wine $\in$ Good-enough S

In all three cases, the fact that categories form a partially ordered set is key to the solution.

For subcategories and optionality,

standard in theories with Typed Feature Structures / constraints (LFG, HPSG, UCG, CCG, etc.) though not in Chomskyan Minimalism

For licensing,

introduced in Bernardi 2002

How is the partial order obtained?

- Stipulated (observed):

  ☹ but better than nothing: still helps with subcategories, optionality, and licensing

or

- Category labels are formulae, ordering given by the derivability relation of some logic:

  patterns of category behavior can be captured using the logic

In either case, logic can determine what feature structures/categories combine (Johnson 1991, Blackburn&Spaan 1993, Doerre&Manandhar 1995, Bernardi 2002, others)

Aside:

Be mindful of the ambiguity of the term “category”:

(i) category label,

(ii) the set of expressions with that label.

Derivability is a notion applicable to labels,

subset relation is applicable to sets of expressions.

(i) vs. (ii) will be disambiguated as needed.

Functional application and subcategories:

If **f** is a function from expressions of category B, then expressions in any subcategory of B are good arguments for **f**.

C is a subcategory of B iff every expression of category C is also of category B (i.e. $C \subseteq B$).

If category labels are formulae of some logic, then derivability among category labels corresponds to the subset relation among the sets of expressions with those labels (Deduction Theorem):

$C \vdash B$ iff $\{e: label(e,C)\} \subseteq \{e: label(e,B)\}$

- Concatenation as functional application

- Scoping also a case of functional application:

  Negation (not that) is of type <t, t> .

  If QP is an expression that denotes a generalized quantifier of type <<e, t>,t>, then QP scopes over a stretch of the sentence H iff H denotes a property of type <e, t>, and QP applies to H: QP(H).

  Recall: λP[every man has P] (λx[a dog bit x])

- Licensing also a case of functional application, specifically, scoping:

Licensor(stretch that contains licensee),

e.g. licensing of NPI:

whether(Mary saw anything)

not_that(Mary saw anything)

Generalizes to licensing of subject/aux inversion, wh-in-situ, etc.:

only_in_that_case(would I meet him)

when(you read which book)

We are going to study

- the fundamentals of NPI-licensing
- the syntax of quantifier scope in Hungarian, where (i) the left-to-right order of quantifiers largely determines relative scope, but (ii) QPs can only line up in particular orders.

(i): scoping in Hungarian = concatenation

(ii): there are restrictions on what sentential subcategory each QP can be quantified into

Slash notation for functor categories:

value-cat/arg-cat vs. arg-cat\value-cat

Every dog	saw	Mary
s/(dp\s)	(dp\s)/dp	dp

Our convention:

Uniformize the categories of QPs and negation-like operators schematically as s_{val}/(dp\\s_{arg}).

Sufficient for the study of

what sentential subcategories the operators take as arguments and return as values.

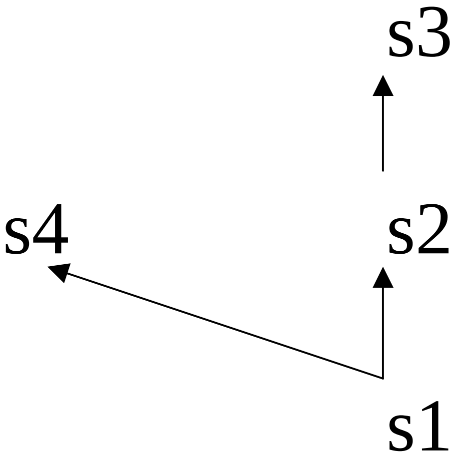
Compare Moortgat's (1991) type constructor **q**

$$\frac{\Gamma \mid\!\text{--}\ \alpha : q(A,B,C) \qquad \Delta[x{:}A] \mid\!\text{--}\ \beta : B}{\Delta[\Gamma] \mid\!\text{--}\ \alpha(\lambda x.\ \beta) : C}\ [qE]$$

An expression of type q(A,B,C) binds a variable of type A, within a domain of type B, producing a meaning recipe of type C.

In our terms, A is the type of **dp**, B is the type of **s_{arg}**, and C is the type of **s_{val}** . Our claims carry over to any grammar that uses **q**.

Let our poset of sentential categories be



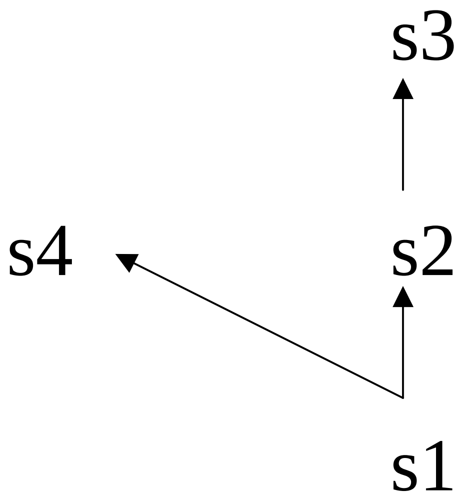
where s3 is the category of complete sentences (derivability arrow to be justified soon)

Optionality of operator OPb:

OPc	**OPb**	OPa	OPc	OPa
s3/s2	s2/s1	s1/s1	s3/s2	s1/s1

because s1 $\longrightarrow$ s2

Let our poset of sentential categories be



where s3 is the category of complete sentences (derivability arrow to be justified soon)

OPe **mediates** between s4 and s3, **not optional**!

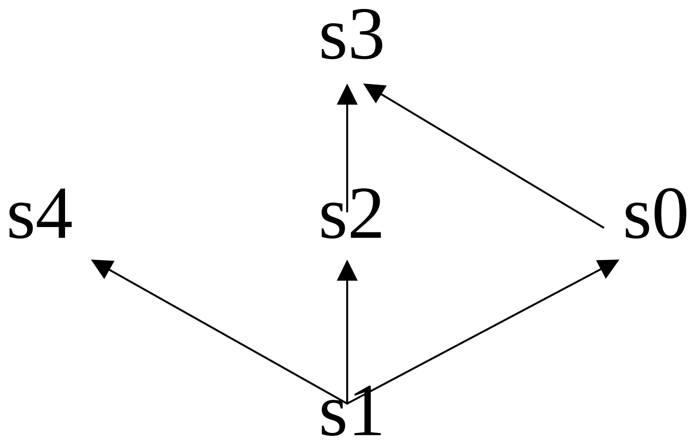
OPf	**OPe**	OPd		* OPf	OPd
s3/s3	s3/s4	s4/s1		s3/s3	s4/s1

because $s4 \not\to s3$

A mediator is a **licensor for an expression E** if E's value category doesn't derive the complete_grammatical_s and the mediator maps it to something that derives that.

OPf	**OPe**	OPd	OPe **licenses** E of s4
s3/s3	s3/s4	s4/s1	

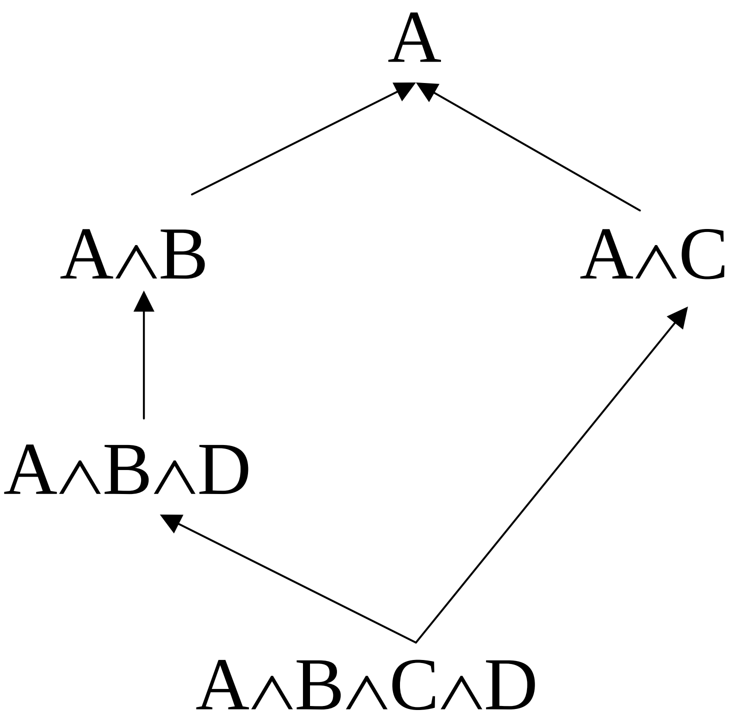
Otherwise a mediator simply makes a particular non-immediate precedence relation possible. E.g. OPh:



OPi	**OPh**	OPg
s0/s0	s0/s2	s2/s2

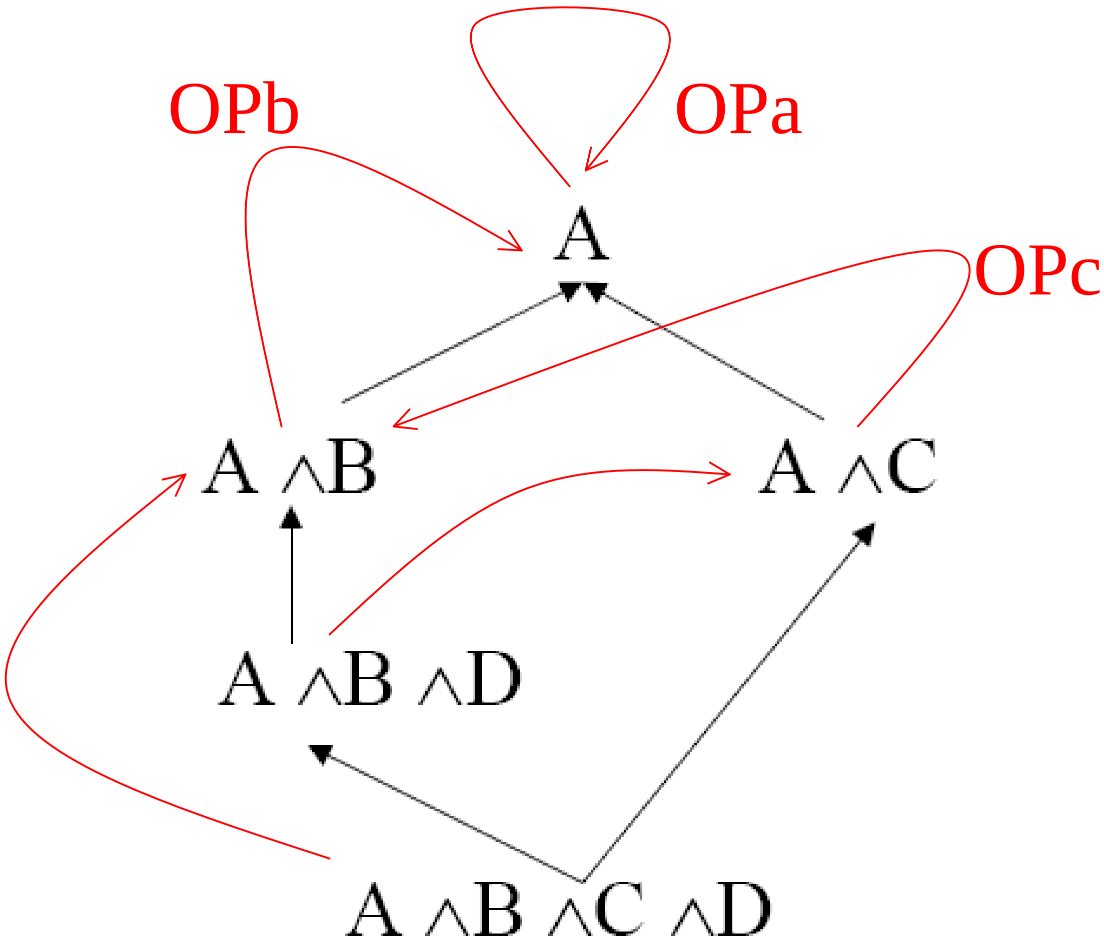
Ordering by derivability: a simple picture

Let ⟨{A, A∧B, A∧C, A∧B∧D, A∧B∧C∧D}, ≤⟩ be a poset, where ≤ is given by **p∧q |-- p.**



(I.e. we have a propositional calculus with just ∧.)

Each atomic proposition says, "I can be immediately preceded by [operator]"



$OPa \in Cat_{A/A}$

$OPb \in Cat_{A/A \wedge B}$

$OPc \in Cat_{A \wedge B/A \wedge C}$

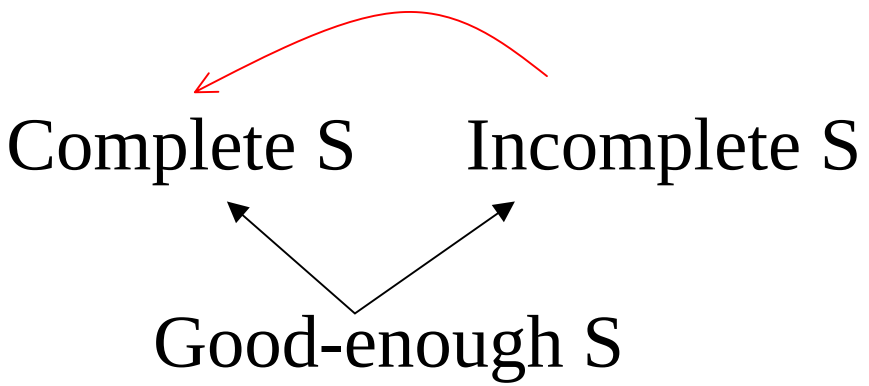
etc.

A = I can be immediately preceded by OPa,

B = I can be immediately preceded by OPb,

C = I can be immediately preceded by OPc,   etc.

Licensing: Recall Bernardi's initial picture



licensor takes Incompl. S & returns Complete S

M drank any more wine ∈ Incompl.S

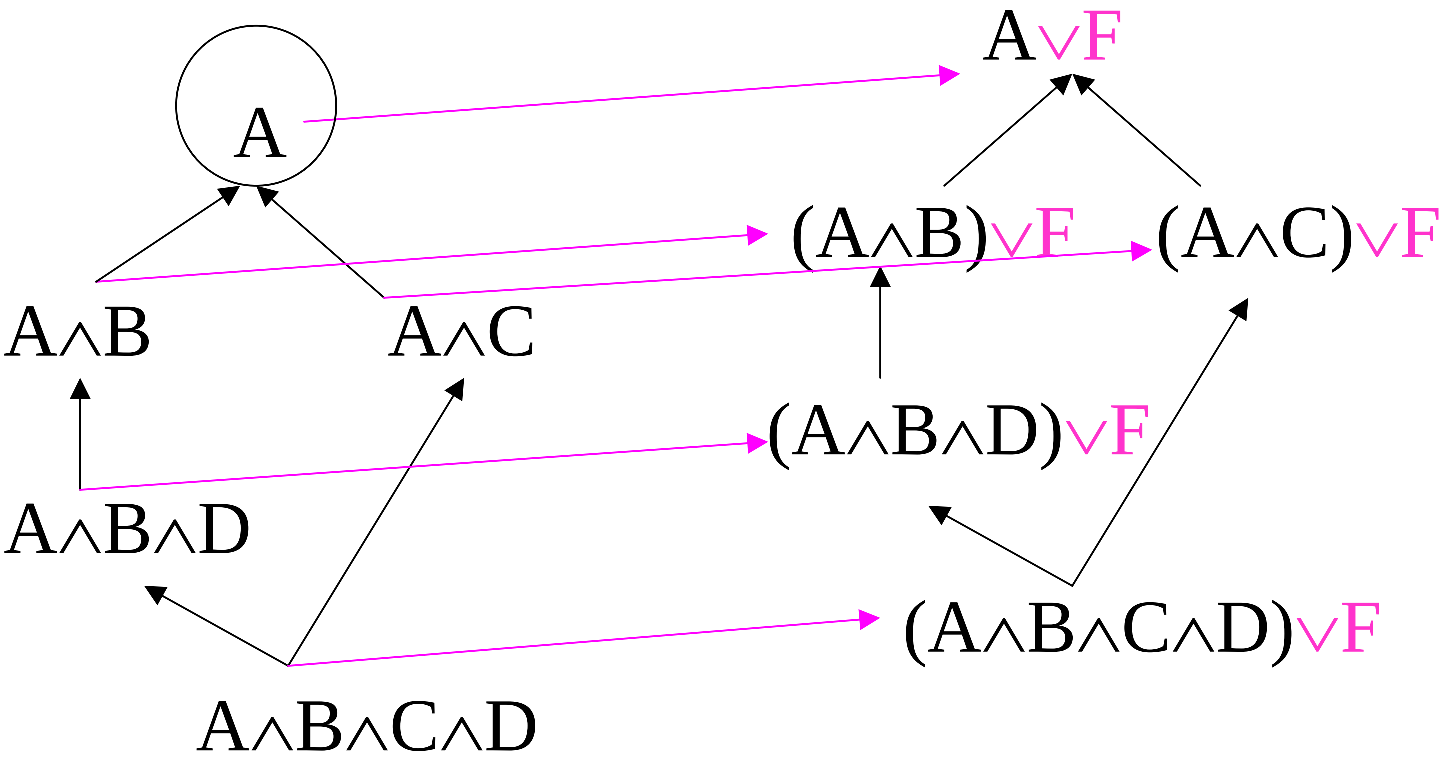
M drank wine ∈ Good-enough S

whether Mary drank (any more) wine ∈ Compl.S

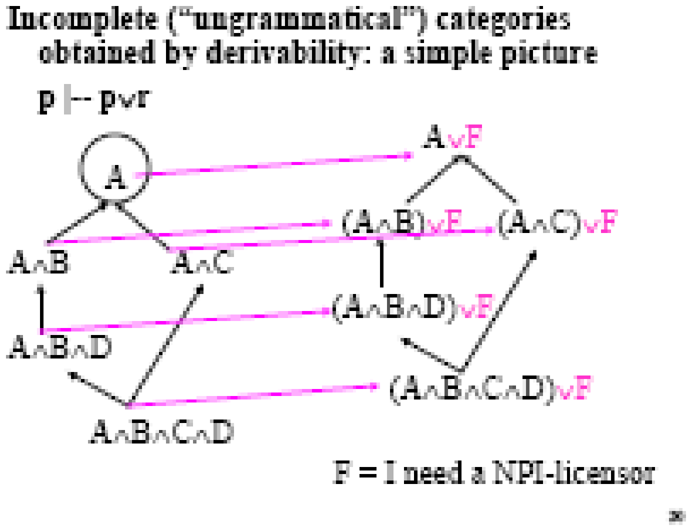
What we want is for Incomplete categories (with unlicensed NPIs) to have the exact same behavior as the corresponding Good-enough ones, save for the presence of the NPI.

Incomplete ("ungrammatical") categories obtained by derivability: a simple picture

p |-- p∨r



F = I need a NPI-licensor



The whole set is partially ordered by derivability, using just ∧ and ∨. All arrows are in fact unidirectional. The "incomplete" layer is isomorphic to the "good-enough" layer. Other incomplete layers can be added as needed.

The use of a logic with $\wedge$ and $\vee$ to define categories is reminiscent of Morrill 1994, Johnson & Bayer 1995, Doerre & Manandhar 1995, but the linguistic application is rather different.

Caveat:

In the CD version of the paper the incomplete ("ungrammatical") categories were handled differently, with conjunction. In that version the good-enough categories had a conjunct "I don't need a NPI-licensor", and derived a corresponding incomplete category that needed a licensor in virtue of not being specified as not needing one. The current version uses an extra connective but linguistically makes more sense. Try not to be confused.

Go to Part 2