Surface Reasoning

Lecture 1: Reasoning with Monotonicity

Thomas Icard

June 18-22, 2012

- This is a course mostly about logical systems designed for, and inspired by, inference in natural language.
- One of the central themes will be that much of the logic of ordinary language can be captured by appealing only to "surface level" features of words and phrases.
- What do we mean by "logic of ordinary language", and what do we mean by "surface level"?

On What Follows from What

- By "logic of ordinary language" we mean what Aristotle had in mind:

  "A deduction is a form of words (logos) in which, certain things having been supposed, something different from those supposed results of necessity because of their being so."
  (Prior Analytics I.2, 24b18-20)

- Basic question: When does one statement follow from another?

- Better: How can we tell when one statement follows from another?
  - How in fact do we, humans, determine whether a statement follows?
  - How could we program a computer to determine this (correctly)?
  - What, even in principle, determines whether a statement follows?

- There seem to be cases where these three subquestions call for different answers. Nonetheless, it is difficult to separate them in theory. This will be a recurring theme.

Surface Information

- "Surface level" reasoning could refer to any number of things: simple, easy, shallow, superficial, tractable, observable, and so on.
- While some of these elements will be present, we mean something more specific.
- We will conceive of languages, whether natural and artificial, as sets of *symbolic structures*, built up from atomic elements by simple rules of combination. Our rules of inference will only be allowed to operate on these symbols: they must operate on the basis of *form* alone.
- Thus, apart from basic relations between symbols, we will be ignoring "deeper" levels of meaning, and we will ignore what might be called "pragmatics" altogether. One might say, we are interested in inferential relations supported directly, or merely, by grammar.
- What is often called *natural logic* is broader than surface reasoning.

Motivating Example

- Consider the following sentence (this example is inspired by [4]):

Most Americans who know a foreign language speak it at home

- Under what conditions is this sentence true? Is it equivalent to:
  - Most Americans who know a foreign language speak at least one of the foreign languages they know at home?
  - Most Americans who know a foreign language speak all the foreign languages they know at home?
  - Most Americans who know a foreign language speak most of the foreign languages they know at home?

- By contrast, the following pattern is patently valid:

$$\frac{\text{Most Americans who know a foreign language speak it at home}}{\text{Most Americans who know a foreign language speak it at home or at work}}$$

- How can this be so easy when assessing truth conditions is so hard?

- One obvious suggestion is that it has something to do with recognizing a generally valid rule, roughly to the effect that:

$$\frac{\text{Most } X\ Y \qquad Y \Rightarrow Z}{\text{Most } X\ Z}$$

- The precise psychological question of how this could work, or whether the underlying psychological mechanism has anything at all to do with such a rule, seems to be open, though at various points, including today, we will discuss some intriguing preliminary work.

- Most of this course will be about logical systems that are designed for this kind of "top-down" deductive strategy. The goal is at least to get a good sense for what information is already there, "on the surface", to be used for inference. This will be illustrated with many concrete examples.

- Much of this work has found applications in computational natural language understanding, and we will discuss some of that as well.

Course Outline

1. The plan for the rest of today is as follows:
   - 1.1 Monotonicity
   - 1.2 Monotonicity and Syllogisms
   - 1.3 Monotonicity in Processing
2. Next, we will delve into the literature on logic-based grammars and grammar-based logics, and discuss several concrete examples:
   - 2.1 Categorial Grammar and Lambek Calculus
   - 2.2 Van Benthem and Sánchez-Valencia's Monotonicity Calculus
   - 2.3 Zamansky et al.'s Order Calculus
3. Then we will dedicate at least one session to negative polarity items and logical systems meant to capture NPI distribution.
4. After that, we will look at inferences that go beyond monotonicity, considering exclusion and other basic relations.
   - 4.1 Extension of Monotonicity Calculus with exclusion relations
   - 4.2 Another connection to NPIs
   - 4.3 MacCartney's NatLog system for RTE
5. Finally, if there is time, we may discuss some similar ideas that have been pursued in the tradition of Quine's Predicate Functor Logic.

Some intriguing aspects of surface reasoning we will not cover include:

- Many entailments follow from grammar or form alone, but have seemingly little to do with logic. Consider phenomena related to the dative alternation (see Levin and Rappaport, among others).
  - Penelope taught Clive archery $\Rightarrow$ Penelope taught archery to Clive
  - Penelope taught Clive archery $\Rightarrow$ Clive learned archery
  - Penelope taught archery to Clive $\not\Rightarrow$ Penelope taught Clive archery
  - Penelope taught archery to Clive $\not\Rightarrow$ Clive learned archery

  One could well imagine developing, axiomatizing, and so on, grammar-based logical systems of the sort we will see, with features that license exactly the right inferences. (C.f. [7].)

- One could also imagine surface reasoning systems that focus more on logical words like 'and', 'or', and so on, as l.u.b. and g.l.b. operators. A number of central entailment relations follow from these properties. This has been explored by a number of researchers (among them Muskens, Zamansky et al., many in proof-theoretic semantics, etc.), but we will not focus on that work here.

- The inference pattern we saw above with 'most' is an example of a *monotonicity inference*. The general form is as follows:

$$\frac{S[X] \qquad X \Rightarrow Y}{S[Y]}\ (\text{mono})$$

The quantifier 'most' is said to be *monotonic* in its second argument, because it supports such an inference.

- By contrast, some quantifiers are *antitonic*, because they support the opposite inference:

$$\frac{S[X] \qquad Y \Rightarrow X}{S[Y]}\ (\text{anti})$$

- For instance, 'all' is antitonic in its first argument:

$$\frac{\text{All sunflowers need sun}}{\text{All Rostov sunflowers need sun}}$$

- In fact every quantifier has a *monotonicity profile*, depending on the monotonicity properties of its argument places:

↓every↑	↑not every↓	*exactly *n**
↑some↑	*most↑	↑at least *n* ↑
↓no↓	*few↓	↓at most *n* ↓

  Here, * means *non-monotone* in that it supports neither "upward" nor "downward" inferences in general.

- These feature are not restricted to quantifiers. Any "functional" expression – verbs, sentential operators, adverbs, adjectives, etc. – can be monotone, antitone, or non-monotone.

- For instance, 'doubt' is antitone, whereas 'believe' is monotone:

$$\frac{\text{He doubted he would win}}{\text{He doubted he would win by 20}} \qquad \frac{\text{He believed he would win by 20}}{\text{He believed he would win}}$$

- This all becomes particularly interesting when we embed monotone/antitone expressions inside others. For instance:

$$\frac{\text{No one doubted he would win by at least 20}}{\text{No one doubted he would win}}$$

- Here the inference gets reversed because we have an antitone context from 'doubted' inside another antitone context from 'no'. This creates a monotone context.

- In general, if we think of antitonic as "negative", $-$, and monotonic as "positive", $+$, then their composition behaves likes negative and positive numbers under multiplication:

$\cdot$	$+$	$-$
$+$	$+$	$-$
$-$	$-$	$+$

- This can in principle be repeated any number of times.

Type Domains

- Recall the standard set $\mathcal{T}$ of types, as the smallest set such that:
  - Basic types $e, t \in \mathcal{T}$.
  - If $\sigma, \tau \in \mathcal{T}$, then $\sigma \rightarrow \tau \in \mathcal{T}$.

- Functional expressions can now be identified as those expressions assigned to functional types. For instance, quantifiers are typically said to be of type $(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$.

- Recall the standard model of type domains $\mathcal{D} = \bigcup_{\tau \in \mathcal{T}} D_\tau$ given by:
  - D_e is assumed to be some fixed set E of entities.
  - $D_t = \{0, 1\}$.
  - $D_{\tau \rightarrow \sigma} = D_\sigma^{D_\tau}$.

  Functional types are so called as they are interpreted as functions.

- (N.B. In what follows we borrow liberally from work by Moss [5].)

Ordered Type Domains

- Notice that $D_t = \{0, 1\}$ is not just a set. It has a simple, natural preordering defined on it: $0 \leq_t 1$.
- We could say that $D_e = E$ has an preordering as well, namely the flat ordering, for which every element is $\leq_e$ to itself, and nothing else is $\leq_e$ to anything else.
- The crucial observation for understanding monotonicity is that functional type domains inherit an order from their argument types (specifically, from that of the co-domain), so these orderings get propagated all the way up the type hierarchy.
- For functions $f, g \in D_{\sigma\to\tau}$ we have:

  $$f \leq_{\sigma\to\tau} g, \text{ if and only if } f(a) \leq_\tau g(a) \text{ for all } a \in D_\sigma.$$

- For predicates this recovers the usual ordering by inclusion.

- Now conceiving of type domains as pairs $\mathbb{D}_\tau = (D_\tau, \leq_\tau)$, we can say formally what it is for a function to be monotone or antitone.

Definition (Monotonicity)

Given preorders $\mathbb{A} = (A, \leq_A)$ and $\mathbb{B} = (B, \leq_B)$, a function $f : A \to B$ is *monotone* just in case, for all $a \in A$:

$$a \leq_A a' \implies f(a) \leq_B f(a').$$

It is *antitone* just in case, for all $a \in A$:

$$a \leq_A a' \implies f(a') \leq_B f(a).$$

It is *non-monotone* if it is neither monotone nor antitone.

- Note that the antitone functions from $\mathbb{A}$ to $\mathbb{B}$ coincide with the monotone functions from $\mathbb{A}$ to $\mathbb{B}^{op} = (B, \geq_B)$. This will become useful in a later lecture.

Examples: Sentential Connectives

- Sentential 'not' is antitone. It is the function $f_\neg$ of type $t \longrightarrow t$ such that $f_\neg(0) = 1$ and $f_\neg(1) = 0$. That is, $v \leq w \Rightarrow f_\neg(w) \leq f_\neg(v)$.

- Sentential 'or' on the other hand is a monotone function $f_\vee$ of type $t \longrightarrow (t \longrightarrow t)$. There are four functions of type $t \longrightarrow t$:

$$\mathbf{0} < \mathtt{id}, f_\neg < \mathbf{1}.$$

  $f_\vee$ sends 0 to $\mathtt{id}$ and 1 to $\mathbf{1}$, hence it is monotone.

- Likewise, $f_\wedge$ sends 0 to $\mathbf{0}$ and 1 to $\mathtt{id}$, so it is also monotone.

- $f_\rightarrow$, the material conditional, is antitone since,

$$f_\rightarrow(1) = \mathtt{id} < \mathbf{1} = f_\rightarrow(0).$$

Currying

- Given preorders $\mathbb{A}$ and $\mathbb{B}$, we can also consider the product preorder $\mathbb{A} \times \mathbb{B}$, for which $(a, b) \leq_{A \times B} (a', b')$ if $a \leq_A a'$ and $b \leq_B b'$.

- Then it is easy to see that:

$$\{f : A \to (B \to C)\} \;=\; \{f : A \times B \to C\}.$$

  Moreover, if we use $[\mathbb{A}, \mathbb{B}]$ to denote the set of monotone functions from A to B, then we also have:

$$[\mathbb{A}, [\mathbb{B}, \mathbb{C}]] \;=\; [\mathbb{A} \times \mathbb{B}, \mathbb{C}].$$

- We can say a function $f : A \times B \to C$ is *monotone in its second argument* if for all $a \in A$, $f(a, \cdot)$ is a monotone from $\mathbb{B}$ to $\mathbb{C}$.

- We can give an analogous definition for *monotone in its first argument*, or indeed *monotone in its n^{th} argument* for any n.

Back to Quantifiers

- The meaning of any quantifier Q is a function

$$q : (E \to 2) \to ((E \to 2) \to 2),$$

or more familiarly,

$$q : \mathcal{P}(E) \to (\mathcal{P}(E) \to 2),$$

and yet more familiarly, in its curried form,

$$q : \mathcal{P}(E) \times \mathcal{P}(E) \to 2.$$

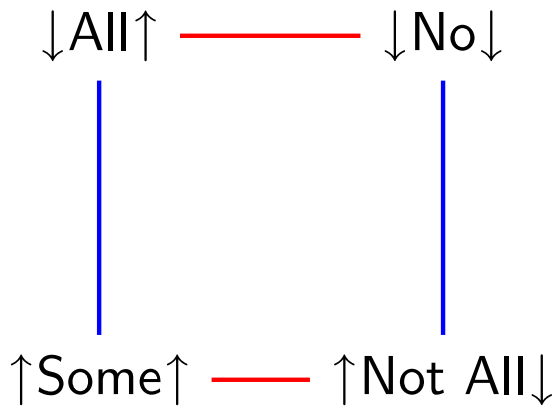
- It is in this sense that 'every' is antitone in its first argument and monotone in its second, that 'no' is antitone in both arguments, 'some' is monotone in both, and so on.

Why Monotonicity?

- Monotonicity principles, as we have seen, are applicable all the way up the type-hierarchy, and they are not particular to any single grammatical category.
- Moreover, there is evidence that such features are in some sense "grammaticalized", a topic we will return to when discussing negative polarity items.
- We will eventually consider features that go beyond monotonicity in interesting ways, but we will first see how much one can already get from these simple, basic principles.
- There are many fundamental inferential principles that one cannot derive from monotonicity. For instance, it does not even give us symmetry of sentential 'and': from 'A and B', derive 'B and A'. An obvious question is, for a given logical system, how far does monotonicity bring us, and how much further do we need to go?

- We can answer this question precisely for the case of the classical syllogism, based on work by van Eijk [2] and Geurts [3].

- First note that the geometry of our monotonicity profile characterizations of the four main quantifiers is captured nicely in the traditional Square of Opposition:



- As van Eijk has shown, every valid syllogism can be derived by exactly one application of monotonicity and at most one application of each of the following rules for each premise or conclusion:

$$\text{(sym)}\ \frac{Q\ A\ B}{Q\ B\ A} \qquad \text{(import)}\ \frac{\text{All}\ A\ B}{\text{Some}\ A\ B} \qquad \frac{\text{No}\ A\ B}{\text{Not All}\ A\ B}$$

- For instance, *barbara* is just a single application of monotonicity:

$$\frac{\text{All } A\ B \qquad \text{All } B\ C}{\text{All } A\ C}\ (\text{mono})$$

- A more interesting example is *fesapo*:

$$(\text{sym})\ \frac{\dfrac{\text{No } C\ B}{\text{No } B\ C} \qquad \dfrac{\dfrac{\text{All } B\ A}{\text{Some } B\ A}\ (\text{import})}{\text{Some } A\ B}\ (\text{sym})}{\text{Some } A\ \overline{C}}\ (\text{mon})$$

- Over the past several years there has been some very interesting work in extended syllogistic logics, most notably by Larry Moss and Ian Pratt-Hartmann. A reasonable question is how the systems we will discuss fit in with that work.

- In general, modern work on syllogistics assumes a restricted syntax and axiomatizes this restricted language over standard models. Often these fragments are small enough to be decidable, and one of the interesting questions is where the line between decidability and undecidability lies.

- In the logical systems we will consider, from the Monotonicity Calculus on, we typically assume unrestricted syntax, but the proof rules never have to be complete for the standard model. The proof rules only pick up on selected semantic features.

- The next theorem suggests some further important differences.

Theorem (van Benthem [1]; Westerståhl [8])

Suppose a quantifier Q is conservative, quantitative, and has extension. Then $Q =$ `every` *if the following rules are valid:*

$$\frac{Q\ A\ B \qquad Q\ B\ A}{A = B} \qquad \frac{}{Q\ A\ A}\ .$$

If Q shows variety, then $Q =$ `every` *if it satisfies the following:*

$$\frac{Q\ A\ B \qquad Q\ B\ C}{Q\ A\ C} \qquad \frac{}{Q\ A\ A}\ .$$

Under the same conditions, $Q =$ `some` *if it satisfies the following rules:*

$$\frac{Q\ A\ B}{Q\ B\ A} \qquad \frac{Q\ A\ B}{Q\ A\ A}\ .$$

- The last topic for this lecture is a paper by Bart Guerts [3], suggesting that these monotonicity features, and related properties, may play some important role in processing of quantifiers.

- The first relevant point, attributed to Oaksford and Chater, is that the following two inference patterns are equally difficult for subjects, despite the difference in logical complexity between 'all' and 'most':

$$\frac{\text{All } A \text{ are } B \qquad \text{All } B \text{ are } C}{\text{All } A \text{ are } C} \qquad \frac{\text{Most } A \text{ are } B \qquad \text{All } B \text{ are } C}{\text{Most } A \text{ are } C}$$

- However, the main goal of the paper is to capture certain trends and phenomena identified in an influential meta-analysis of syllogistic reasoning by Chater and Oaksford.

Recall the figures in the classical syllogism:

- Figure 1:

$$\frac{Q\ B\ C \qquad Q\ A\ B}{Q\ A\ C}$$

- Figure 2:

$$\frac{Q\ C\ B \qquad Q\ A\ B}{Q\ A\ C}$$

- Figure 3:

$$\frac{Q\ B\ C \qquad Q\ B\ A}{Q\ A\ C}$$

- Figure 4:

$$\frac{Q\ C\ B \qquad Q\ B\ A}{Q\ A\ C}$$

premisses	*conclusion*			
& figure	A	I	E	O
AA1	90	5	0	0
AA2	58	8	1	1
AA3	57	29	0	0
AA4	75	16	1	1
AI1	0	92	3	3
AI2	0	57	3	11
AI3	1	89	1	3
AI4	0	71	0	1
IA1	0	72	0	6
IA2	13	49	3	12
IA3	2	85	1	4
IA4	0	91	1	1
AE1	0	3	59	6
AE2	0	0	88	1
AE3	0	1	61	13
AE4	0	3	87	2
EA1	0	1	87	3
EA2	0	0	89	3
EA3	0	0	64	22
EA4	1	3	61	8

premisses	*conclusion*			
& figure	A	I	E	O
AO1	1	6	1	57
AO2	0	6	3	67
AO3	0	10	0	66
AO4	0	5	3	72
OA1	0	3	3	68
OA2	0	11	5	56
OA3	0	15	3	69
OA4	1	3	6	27
II1	0	41	3	4
II2	1	42	3	3
II3	0	24	3	1
II4	0	42	0	1
IE1	1	1	22	16
IE2	0	0	39	30
IE3	0	1	30	33
IE4	0	1	28	44
EI1	0	5	15	66
EI2	1	1	21	52
EI3	0	6	15	48
EI4	0	2	32	27

premisses	*conclusion*			
& figure	A	I	E	O
IO1	3	4	1	30
IO2	1	5	4	37
IO3	0	9	1	29
IO4	0	5	1	44
OI1	4	6	0	35
OI2	0	8	3	35
OI3	1	9	1	31
OI4	3	8	2	29
EE1	0	1	34	1
EE2	3	3	14	3
EE3	0	0	18	3
EE4	0	3	31	1
EO1	1	8	8	23
EO2	0	13	7	11
EO3	0	0	9	28
EO4	0	5	8	12
OE1	1	0	14	5
OE2	0	8	11	16
OE3	0	5	12	18
OE4	0	19	9	14
OO1	1	8	1	22
OO2	0	16	5	10
OO3	1	6	0	15
OO4	1	4	1	25

A = all	E = no
I = some	O = some … not

Important trends:

- People are not bad at syllogisms. They endorse valid syllogisms on average 51% of the time and invalid syllogisms 11% of the time.
- Many errors seem to arise from illicit conversion, e.g. AAnA, $n \neq 1$.
- Geurts is interested in explaining the discrepancies among endorsements of the valid syllogisms. Why are some more difficult than others? Compare, e.g. IA4I and EI4O.

- Geurts proposes a very simplistic processing model, making use of the rules we have seen so far.

- In addition to monotonicity, he considers a rule which was implicit in van Eijk's axiomatization of the syllogism:

$$(\mathrm{N})\ \frac{\text{No } A\ B}{\text{All } A\ \overline{B}}$$

- The proposal is that an abstract reasoner begins a problem with 100 units, and each additional complexity subtracts from this "budget":
  1. For each use of the monotonicity rule subtract 20 units.
  2. For each use of (N) subtract 10 units.
  3. Each time an O-proposition appears in a proof subtract 10 units.

- The intuition behind rule 3 is that the alternative grammatical structure requires some extra processing.

- N.B. In van Eijk's formulation of the syllogism, we would begin with 80, rule 1 would be superfluous, and rule 2 would be changed to subtract 10 when the monotonicity inference is based on an E-proposition assumption.

- The correlation between Geurt's model and Chater and Oaksford's meta-data is surprisingly good ($r = 0.93$):

AA1A	80	(90)	OA3O	70	(69)	EA1O	40	(3)
EA1E	80	(87)	AO2O	70	(67)	EA2O	40	(3)
EA2E	80	(89)	EI1O	60	(66)	EA3O	40	(22)
AE2E	80	(88)	EI2O	60	(52)	EA4O	40	(8)
AE4E	80	(87)	EI3O	60	(48)	AE2O	40	(1)
IA3I	80	(85)	EI4O	60	(27)	AE4O	40	(2)
IA4I	80	(91)	AA1I	60	(5)			
AI1I	80	(92)	AA3I	60	(29)			
AI3I	80	(89)	AA4I	60	(16)			

- These are all the valid syllogisms, where the numbers represent Geurts' scores, with Chater and Oaksford's percentage scores in parentheses.

Summary

- Monotonicity is a central and pervasive feature of "natural reasoning". For this reason, and because it is well understood from a logical point of view, we are taking it as our starting point.
- Monotonicity based reasoning is "close to the surface" in the sense that it does not require full interpretation of the sentences involved.
- On Wednesday we will also see that it is closely linked with important syntactic, or grammatical, phenomena.
- Next time we will look at concrete logical systems, starting from the "parsing as deduction" tradition in formal grammar. This will culminate in a concrete deductive system for monotonicity reasoning.


J. van Benthem. *Essays in Logical Semantics*. Reidel, Dordrecht, 1986.


J. van Eijk. 'Syllogistics = monotonicity + symmetry + existential import', Technical Report SEN-R0512, CWI, Amsterdam, 2005.


B. Geurts. 'Reasoning with Quantifiers', *Cognition*, 86: 231-251, 2003.


B. Geurts and F. van der Slik. 'Monotonicity and Processing Load', *Journal of Semantics*, 22: 97-117, 2005.


L.S. Moss. *Logics for Natural Language Inference*, ESSLLI course notes, 2010.


V. Sánchez-Valencia. *Studies on Natural Logic and Categorial Grammar*. Ph.D. Thesis, University of Amsterdam, 1991.


E. Stabler. 'Natural Logic in Linguistic Theory', LSA 2005 Workshop on *Proof Theory at the Syntax/Semantics Interface*.


D. Westerståhl. 'Some Results on Quantifiers', *Notre Dame Journal of Formal Logic*, 25(2): 152-170.