

Monotonicity

How Determiners Drive Inference

James Pustejovsky

Brandeis University
Department of Computer Science

CS 135: Computational Semantics

Fall 2026 (Session 5A — supplement to Session 5)

Why This Deserves Its Own Session

Session 5 introduced monotonicity in two slides and moved on. It carries more than that.

- It is the **only** property of determiners that directly predicts inference — and inference is what a semantic theory is ultimately for.
- It predicts a **syntactic distribution** (negative polarity items) with no syntactic stipulation.
- It supports a **computational** inference procedure — Natural Logic — that works directly on surface forms, without ever building a logical form.
- And it is the first property we will meet that a distributional representation straightforwardly destroys.

Today: definitions done properly, derivations from denotations, polarity marking, NPI licensing, and where it all breaks.

Today

- ① Two arguments, two monotonicities
- ② The full classification, and how to derive it
- ③ Polarity marking: computing inference on the surface
- ④ Nested contexts and polarity flipping
- ⑤ Negative polarity items
- ⑥ Downward-entailing environments beyond determiners
- ⑦ Natural Logic as a computational system
- ⑧ What monotonicity needs, and what loses it

A Determiner Has Two Arguments

Recall from Session 5: a determiner is a **relation between two sets**.

$$D(Q)(P)$$

$\underbrace{Q}_{\text{restrictor}} \quad \underbrace{P}_{\text{scope}}$

every $\underbrace{\textit{student}}_Q$ $\underbrace{\textit{sleeps}}_P$

So there are two independent questions:

- What happens if we *change the restrictor*, holding the scope fixed?
- What happens if we *change the scope*, holding the restrictor fixed?

Session 5 only looked at the second. Both matter, and for every they go in opposite directions.

Right Monotonicity (the Scope)

Definitions

D is $\text{MON}\uparrow$ (right upward) : $D(Q)(P)$ and $P \subseteq P' \implies D(Q)(P')$

D is $\text{MON}\downarrow$ (right downward) : $D(Q)(P)$ and $P' \subseteq P \implies D(Q)(P')$

In words.

- **Upward:** you may replace the scope by anything *more general*.
- **Downward:** you may replace the scope by anything *more specific*.

Some student **jogs** $\Rightarrow$ Some student **moves** $\text{MON}\uparrow$

No student **moves** $\Rightarrow$ No student **jogs** $\text{MON}\downarrow$

Note that $\text{jog} \subseteq \text{move}$: jogging is a way of moving. The arrows in the two rows point in opposite directions.

Left Monotonicity (the Restrictor)

Definitions

D is $\uparrow$ MON (left upward) : $D(Q)(P)$ and $Q \subseteq Q' \implies D(Q')(P)$

D is $\downarrow$ MON (left downward) : $D(Q)(P)$ and $Q' \subseteq Q \implies D(Q')(P)$

Every student *sleeps* $\Rightarrow$ **Every first-year student** *sleeps* $\downarrow$ MON

Some first-year student *sleeps* $\Rightarrow$ **Some student** *sleeps* $\uparrow$ MON

Notice that every behaves oppositely in its two arguments. Narrowing the restrictor preserves truth; narrowing the scope does not.

That asymmetry is not a curiosity. It is what makes the NPI facts come out right, as we will see.

The Notation

Write the two together, left arrow first:

$\downarrow\text{MON}\uparrow$

means **left downward, right upward** — which is *every*.

Pattern	Restrictor	Scope
$\uparrow\text{MON}\uparrow$	upward	upward
$\downarrow\text{MON}\uparrow$	downward	upward
$\uparrow\text{MON}\downarrow$	upward	downward
$\downarrow\text{MON}\downarrow$	downward	downward

A determiner may also be non-monotone in one or both arguments — neither direction of inference holds. *Exactly two* is the standard example, and we prove it shortly.

The Full Classification

Determiner	Pattern	Condition
<i>every, all</i>	$\downarrow \text{MON} \uparrow$	$Q \subseteq P$
<i>some, a</i>	$\uparrow \text{MON} \uparrow$	$Q \cap P \neq \emptyset$
<i>no</i>	$\downarrow \text{MON} \downarrow$	$Q \cap P = \emptyset$
<i>not every</i>	$\uparrow \text{MON} \downarrow$	$Q \not\subseteq P$
<i>at least two</i>	$\uparrow \text{MON} \uparrow$	$ Q \cap P \geq 2$
<i>at most two</i>	$\downarrow \text{MON} \downarrow$	$ Q \cap P \leq 2$
<i>few</i>	$\downarrow \text{MON} \downarrow$	(roughly) $ Q \cap P $ small
<i>most</i>	non-mon. $\text{MON} \uparrow$	$ Q \cap P > Q - P $
<i>exactly two</i>	non-mon. non-mon.	$ Q \cap P = 2$

Every cell in this table is *derivable* from the condition in the third column. None of it is stipulated — which is the next section.

Deriving: *some* is $\uparrow\text{MON}\uparrow$

$$\llbracket \text{some} \rrbracket(Q)(P) = 1 \iff Q \cap P \neq \emptyset$$

Right upward. Suppose $Q \cap P \neq \emptyset$ and $P \subseteq P'$.

Take $a \in Q \cap P$. Then $a \in Q$, and $a \in P \subseteq P'$, so $a \in P'$.

Hence $a \in Q \cap P'$, so $Q \cap P' \neq \emptyset$.

$\text{MON}\uparrow \checkmark$

Left upward. Suppose $Q \cap P \neq \emptyset$ and $Q \subseteq Q'$.

Take $a \in Q \cap P$. Then $a \in Q \subseteq Q'$ and $a \in P$, so $a \in Q' \cap P$.

$\uparrow\text{MON} \checkmark$

The two proofs are the same proof with the roles swapped — which is exactly what you would expect, since *some* is **symmetric**: $Q \cap P \neq \emptyset$ is symmetric in Q and P .

Deriving: *every* is $\downarrow\text{MON}\uparrow$

$$\llbracket \text{every} \rrbracket(Q)(P) = 1 \iff Q \subseteq P$$

Right upward. Suppose $Q \subseteq P$ and $P \subseteq P'$.

Transitivity of $\subseteq$ gives $Q \subseteq P'$.

MON $\uparrow$ ✓

Left downward. Suppose $Q \subseteq P$ and $Q' \subseteq Q$.

Transitivity again: $Q' \subseteq Q \subseteq P$, so $Q' \subseteq P$.

$\downarrow$ MON ✓

Both directions are transitivity of $\subseteq$, read from the two ends. The subset relation is the whole content of *every*, and monotonicity is just what transitivity looks like when you fix one argument and vary the other.

Deriving: *no* is $\downarrow\text{MON}\downarrow$

$$\llbracket \text{no} \rrbracket(Q)(P) = 1 \iff Q \cap P = \emptyset$$

Right downward. Suppose $Q \cap P = \emptyset$ and $P' \subseteq P$.

Then $Q \cap P' \subseteq Q \cap P = \emptyset$, so $Q \cap P' = \emptyset$.

$\text{MON}\downarrow \checkmark$

Left downward. Symmetric, for the same reason as *some*: the condition is symmetric.

Compare some and no. Their conditions differ only by a negation:

some : $Q \cap P \neq \emptyset$

no : $Q \cap P = \emptyset$

And their monotonicity patterns differ by flipping *both* arrows: $\uparrow\text{MON}\uparrow$ becomes $\downarrow\text{MON}\downarrow$.

Negation reverses monotonicity — a fact we will use heavily in the next section.

Deriving: *exactly two* is Non-Monotone

$$\llbracket \text{exactly two} \rrbracket(Q)(P) = 1 \iff |Q \cap P| = 2$$

To show non-monotonicity we need a **counterexample in each direction**.

Let $\text{student} = \{a, b, c\}$, $\text{jog} = \{a, b\}$, $\text{move} = \{a, b, c\}$, with $\text{jog} \subseteq \text{move}$.

Sentence	$ Q \cap P $	Value
<i>Exactly two students jog</i>	$ \{a, b\} = 2$	true
<i>Exactly two students move</i>	$ \{a, b, c\} = 3$	false

So **not** $\text{MON}\uparrow$. And going the other way, with $\text{sprint} = \{a\}$:

Exactly two students sprint is false ($|\{a\}| = 1$), so **not** $\text{MON}\downarrow$ either.

The reason is structural: *exactly n* is a conjunction of an upward condition ($\geq n$) and a downward one ($\leq n$). Conjoining opposite monotonicities gives none.

Practice: Classify These

Give the monotonicity pattern, and justify with an inference or a counterexample.

- ① *at least three*
- ② *fewer than five*
- ③ *both*
- ④ *neither*

Practice: Classify These

Give the monotonicity pattern, and justify with an inference or a counterexample.

- ① *at least three*
- ② *fewer than five*
- ③ *both*
- ④ *neither*

Answers.

- ① $\uparrow\text{MON}\uparrow$ — $|Q \cap P| \geq 3$; enlarging either set can only grow the intersection.
- ② $\downarrow\text{MON}\downarrow$ — $|Q \cap P| < 5$; shrinking either set can only shrink it.
- ③ $\text{MON}\uparrow$ in scope, and **non-monotone** in restrictor (it presupposes $|Q| = 2$, so changing Q breaks the presupposition).
- ④ $\downarrow\text{MON}\downarrow$ in the same way *no* is, with the same presupposition caveat.

Items 3 and 4 show that presuppositions interact with monotonicity — when changing an argument destroys a presupposition, the inference fails for a reason that has nothing to do with the monotonicity pattern itself.

From Property to Procedure

So far monotonicity is a fact about individual determiners. Now make it a **calculus**.

The idea

Mark every position in a sentence with a **polarity**: $+$ if it occurs in an upward-entailing position, $-$ if downward.

Then: at a $+$ position you may substitute something *more general*; at a $-$ position, something *more specific*.

This turns inference into a **local edit**. No logical form, no theorem prover — just replace one word and check the polarity of its position.

The system is van Benthem's monotonicity calculus, developed by Sánchez Valencia as *Natural Logic*.

Marking a Simple Sentence

Every student⁻ jogs⁺

Every is $\downarrow$ MON $\uparrow$: restrictor gets $-$, scope gets $+$.

Now read off the licensed inferences.

Edit	Position	Result
<i>student</i> $\rightarrow$ <i>first-year student</i>	$-$, so narrow ✓	<i>Every first-year student jogs</i>
<i>jogs</i> $\rightarrow$ <i>moves</i>	$+$, so widen ✓	<i>Every student moves</i>
<i>student</i> $\rightarrow$ <i>person</i>	$-$, widening ✗	<i>not licensed</i>
<i>jogs</i> $\rightarrow$ <i>sprints</i>	$+$, narrowing ✗	<i>not licensed</i>

And edits compose: *Every first-year student moves* follows, by making both licensed edits at once.

Marking with *no*

No student⁻ moves⁻

No is $\downarrow\text{MON}\downarrow$, so **both** arguments are negative.

<i>No student moves</i> $\Rightarrow$ <i>No first-year student moves</i>	narrow the restrictor
<i>No student moves</i> $\Rightarrow$ <i>No student jogs</i>	narrow the scope
<i>No student moves</i> $\Rightarrow$ <i>No first-year student jogs</i>	both at once

Compare with *some*, which is $\uparrow\text{MON}\uparrow$ — both positions positive, and every inference runs the other way.

One determiner choice flips the entire inferential behaviour of the sentence. That is a lot of work for one word, and it is why the classification table earns its keep.

Nested Contexts: Polarity Flips

Polarity is computed by **composing** the operators a position sits under.

The rule

Start at $+$ at the root. Each downward-entailing operator you pass through **flips** the polarity.

One flip. *No student who owns a car⁻ is happy*

The restrictor of *no* is DE, so narrowing *car* to *sports car* is licensed.

Two flips. *Every student who doesn't own a car⁺ is happy*

Operator	Effect	Running polarity
—	start	+
restrictor of <i>every</i>	flip	—
<i>doesn't</i>	flip	+

Check: widening *car* to *vehicle* gives *every student who doesn't own a vehicle is happy*.

Non-vehicle-owners are a subset of non-car-owners, so the restrictor narrowed — licensed. ✓

Double Negation

No student failed to move⁺

Two downward operators stack:

Operator	Effect	Running polarity
—	start	+
<i>no</i> (scope)	flip	—
<i>failed to</i>	flip	+

So the inference runs upward: *No student failed to jog* $\Rightarrow$ *No student failed to move*

Correct — it says every student jogged, hence every student moved.

Polarity composes multiplicatively: two negatives make a positive, exactly as in arithmetic.
An even number of DE operators restores upward entailment.

The Data

Some expressions occur only in certain environments.

	Sentence	Environment
ok	<i>No student had <u>any</u> money</i>	scope of <i>no</i>
*	<i>Some student had <u>any</u> money</i>	scope of <i>some</i>
*	<i>Every student had <u>any</u> money</i>	scope of <i>every</i>
ok	<i>Every student who had <u>any</u> money left</i>	restrictor of <i>every</i>
ok	<i>Alice didn't have <u>any</u> money</i>	under negation
ok	<i>Few students had <u>any</u> money</i>	scope of <i>few</i>

Any, ever, yet, lift a finger, budge an inch all pattern alike.

Six environments, and the acceptable ones look unrelated at first glance. What do *no*, the restrictor of *every*, negation, and *few* have in common?

Ladusaw's Generalization

Ladusaw (1979)

Negative polarity items are licensed exactly in **downward-entailing** environments.

Check it against the data:

Environment	Polarity	NPI?
scope of <i>no</i>	—	ok
scope of <i>some</i>	+	*
scope of <i>every</i>	+	*
restrictor of <i>every</i>	—	ok
under negation	—	ok
scope of <i>few</i>	—	ok

Perfect correspondence. A *semantic* property — how a position behaves under set inclusion — predicts a *syntactic* fact about where a word may appear.

Note the crucial case: *every* licenses NPIs in one argument and not the other. No account treating *every* as uniformly positive or negative could get that right.

Why It Might Be True

The generalization is robust. Why would language work this way?

One influential answer (Kadmon & Landman; Krifka; Chierchia). NPIs like *any* are **domain wideners**: they signal that even the most marginal cases are included.

- In a **downward** environment, widening the domain makes the statement *stronger*. *I don't have any money* is stronger than *I don't have money*.
- In an **upward** environment, widening makes it *weaker* — and asserting a gratuitously weakened claim is odd.

So the licensing condition falls out of a pragmatic preference for strength, given the monotonicity of the position.

This is a good example of the division of labour in semantics: the *distribution* is predicted by monotonicity, and the *reason* for it is pragmatic.

Where the Generalization Strains

Only licenses **NPIs** — but is not downward entailing in the relevant argument.

Only Alice had any money. ok

And the restrictor of every is DE only under a presupposition. Consider:

Every student left $\Rightarrow$ Every student with a hat left

Fine — unless the inference requires that there *be* students with hats.

Von Stechow's repair: Strawson downward entailment. Require the inference to go through only when the conclusion's presuppositions are satisfied. Under that weakening, *only* is Strawson-DE and the generalization survives.

Worth seeing: a clean generalization, a genuine counterexample, and a principled weakening that rescues it. That is how this field actually progresses.

Downward-Entailing Environments

Determiners are not the only source of negative polarity.

Environment	Example	NPI ok?
sentential negation	<i>Alice didn't <u>ever</u> call</i>	yes
antecedent of a conditional	<i>If you have <u>any</u> questions, ask</i>	yes
before-clause	<i>She left before <u>anyone</u> arrived</i>	yes
without	<i>He left without <u>any</u> warning</i>	yes
comparative	<i>Taller than <u>anyone</u> here</i>	yes
rarely, seldom	<i>She rarely says <u>anything</u></i>	yes
after-clause	<i>*She left after <u>anyone</u> arrived</i>	no
often	<i>*She often says <u>anything</u></i>	no

Check the DE claim on *before*: *She left before anyone arrived* entails *She left before any student arrived* — narrowing the scope preserves truth. *After* does not work this way, and correspondingly blocks NPIs.

A Minimal Pair Worth Knowing

Few students passed. vs. *A few students passed.*

	Monotonicity	NPI
<i>few</i>	↓MON↓	<i>Few students said <u>anything</u></i> ok
<i>a few</i>	↑MON↑	<i>*A few students said <u>anything</u></i> *

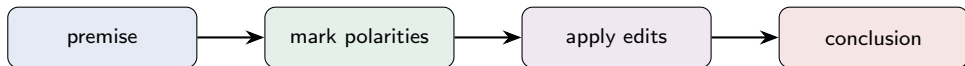
One indefinite article, and the entire inferential profile inverts.

This is the cleanest available demonstration that NPI licensing tracks *monotonicity* rather than anything about negation, form, or lexical marking. The two expressions are nearly identical in shape and meaning; only the direction of entailment differs.

Test it yourself: *Few students moved* $\Rightarrow$ *Few students jogged* ✓; *A few students jogged* $\Rightarrow$ *A few students moved* ✓. Opposite directions.

Inference Without Logical Forms

Everything so far suggests a computational procedure.



Natural Logic (Sánchez Valencia; MacCartney & Manning) does inference by editing surface forms, using a lexical resource such as WordNet to know which words are more general than which.

No logical form is ever built	no parsing to formulas, no theorem prover
Edits are local	one word at a time, composable
Limited expressiveness	no arithmetic, no long chains of reasoning

The trade: much less power than first-order inference, at a tiny fraction of the cost — and it covers a surprising share of the entailments in natural text.

A Natural Logic Derivation

Premise: *No student who owns a car commutes by bus.*

Question: does it entail *No first-year student who owns a sports car commutes by school bus?*

Edit	Direction	Position	Licensed?
<i>student</i> → <i>first-year student</i>	narrow	restrictor of <i>no</i> , —	yes
<i>car</i> → <i>sports car</i>	narrow	inside restrictor, —	yes
<i>bus</i> → <i>school bus</i>	narrow	scope of <i>no</i> , —	yes

All three licensed — the entailment holds. Every position in this sentence is negative, because *no* is $\downarrow\text{MON}\downarrow$ and both of its arguments therefore narrow.

Now change one edit. Replace the second with *car* → *vehicle* — a *widening* at a — position.

Not licensed, and the counterexample is easy: a student who owns a motorcycle but no car, and commutes by bus. The premise is silent about her; the conclusion is not.

Monotonicity Requires an Order

Look at what every definition today had in common.

$$P \subseteq P' \implies \dots$$

Every statement of monotonicity is a claim about $\subseteq$. The sets of individuals form a **partial order** under inclusion, and a determiner is a map between orders that either preserves or reverses it.

Monotone increasing	order-preserving
Monotone decreasing	order-reversing
Non-monotone	neither

This is the same notion of monotonicity as in analysis or order theory. Nothing about it is linguistic — what is linguistic is the empirical claim that determiner meanings are almost always monotone in each argument.

And Orders Are Exactly What Vectors Lose

Session 6B encodes a predicate as an indicator vector $\mathbf{p} \in \{0, 1\}^n$.

	Sets	Indicator vectors
Order	$P \subseteq P'$	$\mathbf{p} \leq \mathbf{p}'$ componentwise
Monotonicity	well defined	well defined

In the *exact regime*, everything survives. Componentwise $\leq$ on indicator vectors is exactly $\subseteq$, so today's results transfer unchanged.

In the approximate regime, it does not. Dense embeddings have no natural partial order — $\mathbf{v}_{\text{jog}}$ is not “ $\leq$ ” $\mathbf{v}_{\text{move}}$ in any sense the geometry provides. And with no order, there is no monotonicity, and therefore no natural-logic inference.

So the loss we described abstractly in Session 6B has a concrete casualty: this entire session's inference machinery.

Case Study: *few* vs. *a few*

Take the minimal pair from earlier and ask what each framework does with it.

In the exact encoding (Session 6B), with **a** the restrictor and **b** the scope:

$$\text{few}(Q)(P) \iff \mathbf{a} \cdot \mathbf{b} < k \qquad \mathbf{a} \text{ few}(Q)(P) \iff \mathbf{a} \cdot \mathbf{b} \geq k$$

		Condition	Shrink either set
<i>few</i>	$\mathbf{a} \cdot \mathbf{b} < k$	intersection shrinks, still $< k$	$\downarrow \text{MON} \downarrow$
<i>a few</i>	$\mathbf{a} \cdot \mathbf{b} \geq k$	needs growth, not shrinkage	$\uparrow \text{MON} \uparrow$

Both compute the same inner product. They differ only in the comparison operator: $<$ versus $\geq$.

One bit, in the decision boundary — and the entire inferential profile follows from it.

Why Dense Distributions Fail Here

Three independent problems, any one of which is fatal.

- ① **Distributional near-identity.** *Few* and *a few* occur in the same slots, with the same nouns, in the same registers. Their co-occurrence profiles are about as similar as two English expressions get.

The distributional hypothesis therefore predicts they mean nearly the same thing — and it is confidently wrong.

- ② **The distinguishing token is a stopword.** The whole contrast is carried by *a*. Any pipeline with stopword removal makes the two expressions *literally identical* — standard practice in distributional semantics for decades.

- ③ **No order, so nowhere to put the flip.** Even granting the model distinguishes them, $\downarrow\text{MON}\downarrow$ versus $\uparrow\text{MON}\uparrow$ is a fact about how an operator *transforms* $\subseteq$. With no partial order on meanings, there is no structure for “reverses the order” to be a property of.

Note that (1) is worse than having no opinion: the framework makes a prediction, and the prediction is inverted.

Can a Transformer Do It?

In principle, yes. Contextual embeddings compose *a* and *few* into a distinct representation, so the two phrases are not forced to be identical the way a bag-of-words model forces them to be.

But notice what has to be learned. Not that they are different words — that their **entailment direction is opposite**. That is a functional property, acquirable only from enough downward- versus upward-entailing contrasts.

Working against it	downward-entailing contexts are comparatively rare in text the cue is one high-frequency function word the correct inference <i>reverses</i> , rather than weakening
---------------------------	--

Empirically this pair appears in monotonicity NLI datasets — HELP, MED — precisely because it is handled unevenly, with the downward direction weaker.

Which is the honest verdict: not impossible, not reliable, and expensive to get right.

What Would Actually Be Required

- ① **A partial order on predicate meanings.** *Solved* — order embeddings, box embeddings, hyperbolic embeddings all supply one.
- ② **Determiners as monotone maps between ordered spaces**, with a *provable* direction in each argument. *Open*.
- ③ **A mechanism for the flip.** *Order-reversal is discrete* — a sign change — and continuous machinery does not produce discrete decisions naturally.

Compare what the symbolic account needed: **a single comparison operator**. The difference between $<$ and $\geq$ is not something a continuous representation has a natural home for — and that, rather than any lack of expressive power, is why this minimal pair is hard.

It is also a good test to run yourself. Take any embedding model, and check whether it distinguishes the two in a way that supports the right inferences. The answer is informative either way.

An Open Question

Can a distributional representation support monotonicity?

Attempts exist.

- **Order embeddings** (Vendrov et al.): embed so that $\mathbf{p} \leq \mathbf{p}'$ componentwise encodes hyponymy. Imposes an order by construction.
- **Box embeddings**: represent concepts as axis-aligned regions, with containment as the order.
- **Hyperbolic embeddings**: exploit the tree-like geometry of hyperbolic space to encode hierarchy.

All three work by **putting the order back in** — choosing a geometry in which inclusion has a meaning. None of them gets it for free.

A candidate final project: take a monotonicity-marked inference dataset, and measure whether any of these representations actually supports the entailments that today's calculus licenses. To my knowledge that has not been done cleanly.

Summary

Concept	Key point
Two arguments	restrictor and scope, monotone independently
Notation	$\downarrow\text{MON}\uparrow$ — left arrow first
<i>every</i>	$\downarrow\text{MON}\uparrow$; both directions are transitivity of $\subseteq$
<i>some</i>	$\uparrow\text{MON}\uparrow$; <i>no</i> is $\downarrow\text{MON}\downarrow$
<i>exactly two</i>	non-monotone; a conjunction of opposite conditions
Polarity marking	+ widen, – narrow; computed by composition
Flipping	each DE operator flips; two flips restore +
Ladusaw	NPIs licensed exactly in DE environments
<i>few</i> vs <i>a few</i>	minimal pair; monotonicity inverts, NPI follows
Strawson-DE	von Fintel's repair for <i>only</i> and presupposition
Natural Logic	inference by surface edits; no logical form
The requirement	monotonicity needs a partial order on meanings

Reading

Core:

- Barwise & Cooper (1981), *Generalized Quantifiers and Natural Language* — monotonicity introduced
- Ladusaw (1979), *Polarity Sensitivity as Inherent Scope Relations*
- van Benthem (1986), *Essays in Logical Semantics* — the monotonicity calculus

Computational:

- MacCartney & Manning (2007, 2009) — Natural Logic for textual inference
- Bowman et al. and successors on whether neural models learn monotonicity

Refinements:

- von Fintel (1999), *NPI Licensing, Strawson Entailment, and Context Dependency*
- Chierchia (2013), *Logic in Grammar* — the pragmatic derivation

Exercises. Classify *at most three, between two and five, not all*. Mark polarities in *Nobody who didn't bring any money was admitted*. Then check your marking against the entailments.

Questions?

Back to the main track — Session 6