Can Computation Give Rise to Meaning?

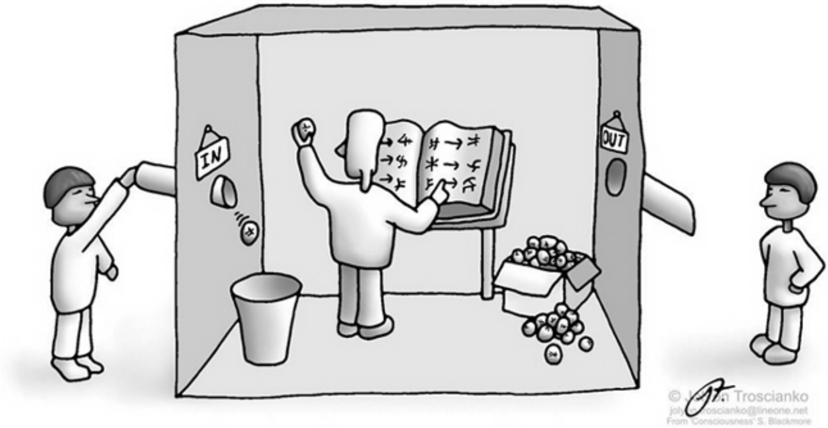
Cem Bozşahin
Cognitive Science Department
Middle East Technical University (ODTÜ)
bozsahin@metu.edu.tr

Boğaziçi University, Philosophy Department Seminars

May 5, 2014

- Turing (1950): Check it out.
  - Semantics as verbal behavior
- Searle (1980): No.
  - Computation is purely formal (syntactic)
  - We need the right stuff (brain) to cause semantics
- Rapaport (1986): Yes.
  - Syntactic semantics (tripartite compositional semantics)
    $2x + 4 = 5$ has syntactical semantics and "physical" semantics (tripartite relation)
  - This is not a wetware/hardware problem
  - Program qua algorithm does not understand, but the running process does

Searle-in-the-box: Chinese Room



The original thought experiment is Rogers (1959)

- CR is an ill-thought experiment: No grammar can be turned into a look-up table of forms and fit into a finite-size room. Bozsahin (2006, 2012)

  Rey (1986); Rapaport (2006)
- If humans only exchanged forms like CR, they could not learn meanings either
- They appear to triangulate forms with some verbal or bodily behavior and interaction

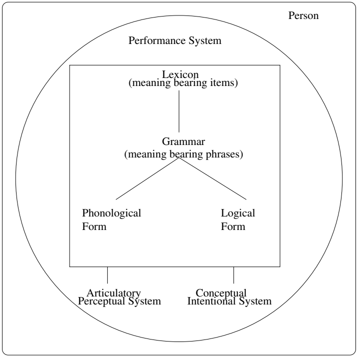
Two channels in sensory impairments or experiential deprivation

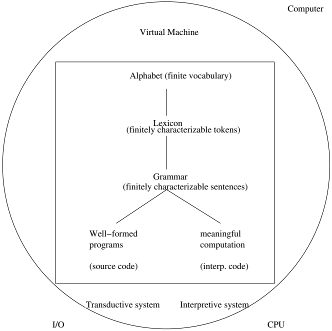
- Did Helen Keller learn to cause semantics? Rapaport (2006)

  How Helen Keller used syntactic semantics to escape from a Chinese Room
- Two channels of information:
  - forms
  - behavior, action, and observation (of the world, internal and external)

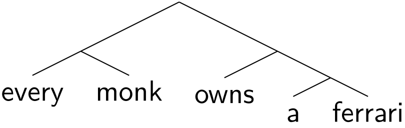
- Children with autism manifest soliloquy
- Blind children learn the difference between look and watch
  Landau and Gleitman (1985): you can touch the table but don't look at it!
- Deaf children acquire sign language (not gestures)
  If they are exposed to data in the critical period, just like other children

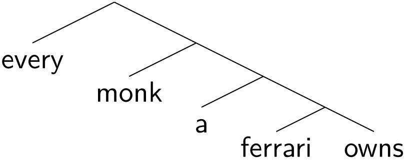
Grammar, cognition and computation: bird's eye view





Logician's view of: every monk owns a ferrari



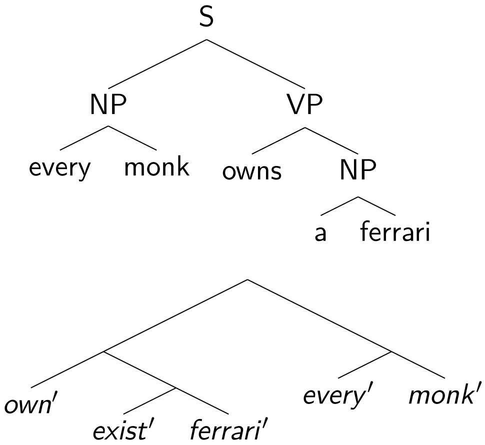


$$(\forall x)(\mathit{monk}(x) \rightarrow (\exists y)(\mathit{ferrari}(y) \wedge \mathit{owns}(x, y)))$$

$$(\exists y)(\mathit{ferrari}(y) \wedge (\forall x)(\mathit{monk}(x) \rightarrow \mathit{owns}(x, y)))$$

Linguist's view

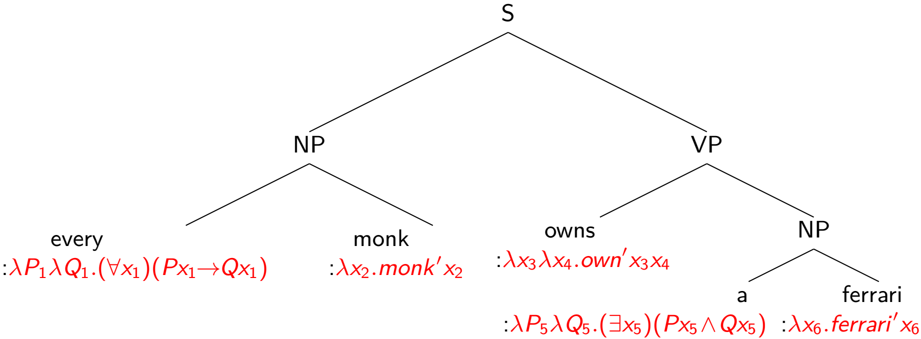
Heads of substantive phrases have lexical content



What are primes about?

Computational Linguist's view

All head dependencies are efficiently computable

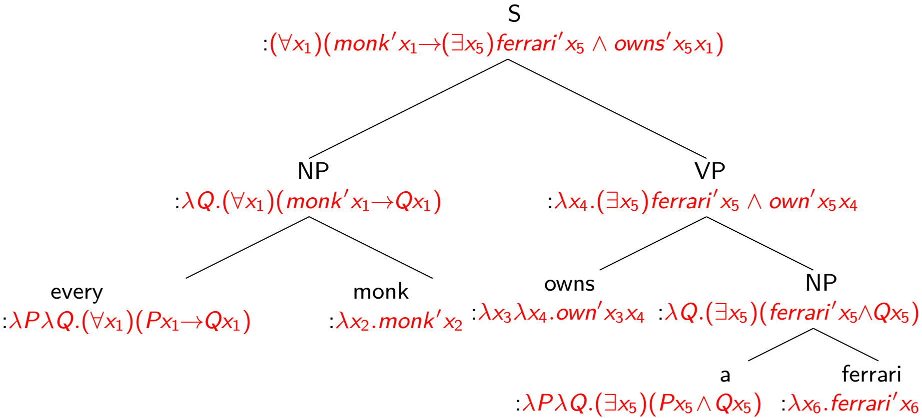


Where do primes come from?

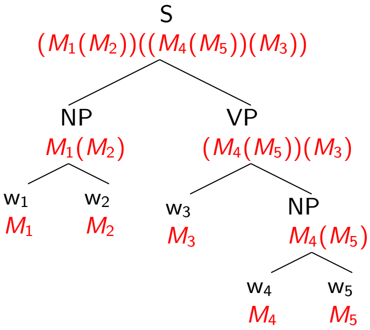
What causes semantics?

How does a word come to be about a prime?

Projecting structure



Meaning without lexical content of words



Lexical content cannot be predicted from grammar

Meaning without grammar

every
$:\lambda P \lambda Q.(\forall x_1)(P x_1 \rightarrow Q x_1)$

monk
$:\lambda x_2.monk' x_2$

owns
$:\lambda x_3 \lambda x_4.own' x_3 x_4$

a
$:\lambda P \lambda Q.(\exists x_5)(P x_5 \wedge Q x_5)$

ferrari
$:\lambda x_6.ferrari' x_6$

Structure of meaning cannot be predicted without a grammar

Meaning from word-grammar

S	$\rightarrow$	NP VP	$S' = NP'(VP')$
VP	$\rightarrow$	V NP	$VP' = NP'(V')$
NP	$\rightarrow$	Det N	$NP' = Det'(NP')$
Det	$\rightarrow$	every	$Det' = \lambda P \lambda Q.(\forall x)(Px \rightarrow Qx)$
Det	$\rightarrow$	a	$Det' = \lambda P \lambda Q.(\exists x)(Px \wedge Qx)$
N	$\rightarrow$	monk	$N' = \lambda x.monk'x$
N	$\rightarrow$	ferrari	$N' = \lambda x.ferrari'x$
V	$\rightarrow$	owns	$V' = \lambda x \lambda y.owns'xy$

Top part projects, and bottom part initiates meaning (hence the dichotomy)

Can we predict structure and lexical content together?

A causal mechanism for expressible/expressed meanings

Reducing a grammar to its lexicon without loss of structure

Every right-hand side has one symbol; such rules are functions looking from constituent's perspective

S	$\to$	NP VP	$S' = NP'(VP')$	$NP{=}S/VP$	$VP{=}S\backslash NP$
VP	$\to$	V NP	$VP' = NP'(V')$	$V{=}VP/NP$	$NP{=}VP\backslash V$
NP	$\to$	Det N	$NP' = Det'(NP')$	$Det{=}NP/N$	$N{=}NP\backslash Det$

Slashed cats: structure-equivalent combinatory categories (eqv. under substitution)

$NP{=}S/(S\backslash NP)$ $\quad$ $V{=}(S\backslash NP)/NP$ $\quad$ $NP{=}(S\backslash NP)\backslash((S\backslash NP)/NP)$

$Det{=}(S/(S\backslash NP))/N$ $\quad$ $Det{=}((S\backslash NP)\backslash((S\backslash NP)/NP))/N$

$N{=}(S/(S\backslash NP))\backslash((S/(S\backslash NP))/N)$
$N{=}((S\backslash NP)\backslash((S\backslash NP)/NP))\backslash((S\backslash NP)\backslash((S\backslash NP)/NP)/N)$

We've got N, V, Det without a need for NP, VP or S rule

N, V, Det are the only lexical categories in the grammar!

$$\begin{array}{lll} \text{S} & \rightarrow & \text{NP VP} \\ \text{VP} & \rightarrow & \text{V NP} \\ \text{NP} & \rightarrow & \text{Det N} \end{array}$$

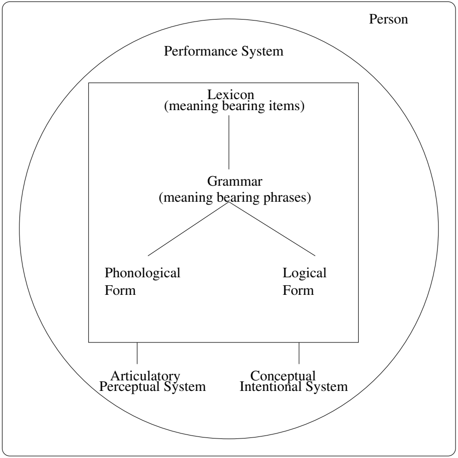
$$\begin{array}{lll} \text{Det} & \rightarrow & \text{every} \\ \text{Det} & \rightarrow & \text{a} \\ \text{N} & \rightarrow & \text{monk} \\ \text{N} & \rightarrow & \text{ferrari} \\ \text{V} & \rightarrow & \text{owns} \end{array}$$

Two grammars capture the same structures and meanings:

S	$\rightarrow$	NP VP	$S' = NP'(VP')$
VP	$\rightarrow$	V NP	$VP' = NP'(V')$
NP	$\rightarrow$	Det N	$NP' = Det'(NP')$
Det	$\rightarrow$	every	$Det' = \lambda P \lambda Q.(\forall x)(Px \rightarrow Qx)$
Det	$\rightarrow$	a	$Det' = \lambda P \lambda Q.(\exists x)(Px \wedge Qx)$
N	$\rightarrow$	monk	$N' = \lambda x.monk'x$
N	$\rightarrow$	ferrari	$N' = \lambda x.ferrari'x$
V	$\rightarrow$	owns	$V' = \lambda x \lambda y.owns'xy$

every	:=	$(S/(S\backslash NP))/N$	: $\lambda P \lambda Q.(\forall x)(Px \rightarrow Qx)$
a	:=	$((S\backslash NP)\backslash((S\backslash NP)/NP))/N$	: $\lambda P \lambda Q.(\exists x)(Px \wedge Qx)$
monk	:=	N	: $\lambda x.monk'x$
ferrari	:=	N	: $\lambda x.ferrari'x$
owns	:=	$(S\backslash NP)/NP$	: $\lambda x \lambda y.owns'xy$

Difference: in the red corner, form-meaning relation only through words



$$\text{owns} := (S\backslash NP)/NP : \lambda x \lambda y.owns'xy$$

Real semantics arising from computation is probably not

the proxy objects like $monk'$, $ferrari'$, $every'$

But the process of their construction.

Learning veggies are veggies, eating is eating, plural is plural

Eat veggies.

possible hypotheses:

$eat{:=}S/NP{:}eat'$	$veggies{:=}NP{:}veg'$	
$eat{:=}S/NP{:}eat'$	$veggie{:=}NP{:}veg'$	$\text{-}s := NP\backslash NP{:}plu'$
$eat{:=}NP{:}eat'$	$veggies{:=}S\backslash NP{:}\ \lambda x.veg'x$	
$eat{:=}NP{:}veg'$	$veggies{:=}S\backslash NP{:}\ \lambda x.eat'x$	
$eat{:=}S/NP{:}eat'$	$veggie{:=}NP/NP{:}plu'$	$\text{-}s :=\ NP{:}veg'$

impossible hypotheses:

$*eat{:=}NP{:}eat'$	$veggies{:=}\ S/NP{:}\ veg'$	
$*eat{:=}S\backslash NP{:}eat'$	$veggies{:=}\ NP{:}\ veg'$	
$*eat{:=}S\backslash NP{:}eat'$	$veggie{:=}NP{:}veg'$	$\text{-}s := NP\backslash NP{:}plu'$

No veggies.

Assume: chocolate around

Experience 1 (Eat veggies)

eat $:= S/NP : eat'$
$\quad : veg'$
$NP : eat'$
$\quad : veg'$

veggies $:= S\backslash NP : veg'$
$\quad : eat'$
$\quad : plu'\ veg'$
$\quad : plu'\ eat'$
$NP : veg'$
$\quad : eat'$
$\quad : plu'\ veg'$
$\quad : plu'\ eat'$

veggie $:= NP : veg'$
$NP/NP : plu'$
$\quad : veg'$

-s $:= NP\backslash NP : plu'$
$NP : veg'$

Experience 2 (No veggies; with chocolate)

no $:= S/NP : no'$
$\quad : veg'$
$\quad : choc'$

veggies $:= S\backslash NP : no'$
$\quad : veg'$
$\quad : choc'$
$\quad : eat'$
$\quad : plu'\ veg'$
$\quad : plu'\ choc'$
$\quad : plu'\ no'$
$NP : veg'$
$\quad : eat'$
$\quad : no'$
$\quad : choc'$
$\quad : plu'\ veg'$
$\quad : plu'\ choc'$
$\quad : plu'\ no'$

veggie $:= NP : no'$
$\quad : veg'$
$\quad : choc'$
$NP/NP : plu'$
$\quad : veg'$
$\quad : choc'$

-s $:= NP\backslash NP : plu'$
$NP : veg'$
$\quad : choc'$

Even in this circumscribed world of two experiences only, the child is exponentially less likely to believe that veggies could mean negation, eating, plural or chocolate, rather than veggies.

Veggies gone.

$$
\begin{array}{lll}
\text{veggies} := & S/NP & : veg' \\
& & : gone' \\
& & : eat' \\
& & : no' \\
& & : plu'\, veg' \\
& NP & : veg' \\
& & : gone' \\
& & : eat' \\
& & : no' \\
& & : plu'\, gone'
\end{array}
\quad
\begin{array}{lll}
\text{gone} := & S\backslash NP & : veg' \\
& & : gone' \\
& NP & : veg' \\
& & : gone'
\end{array}
\quad
\begin{array}{lll}
\text{veggie} := & NP & : veg' \\
& & : no' \\
& NP/NP & : veg' \\
& & : plu'
\end{array}
\quad
\begin{array}{lll}
\text{-s} := & NP\backslash NP & : plu' \\
& NP & : veg' \\
& S/NP & : gone' \\
& & : veg' \\
& & : plu'
\end{array}
$$

$$
\{\text{veggies, veggie}\} := \{
\begin{array}{lll}
S\backslash NP{:}veg' @ \frac{2}{55}, & S\backslash NP{:}eat' @ \frac{2}{55}, & S\backslash NP{:}no' @ \frac{1}{55}, \\
S\backslash NP{:}choc' @ \frac{1}{55}, & S\backslash NP{:}plu'\,veg' @ \frac{2}{55}, & S\backslash NP{:}plu'\,eat' @ \frac{1}{55}, \\
S\backslash NP{:}plu'\,no' @ \frac{1}{55}, & S\backslash NP{:}plu'\,choc' @ \frac{1}{55}, & \\
S/NP{:}veg' @ \frac{2}{55}, & S/NP{:}gone' @ \frac{2}{55}, & S/NP{:}eat' @ \frac{1}{55}, \\
S/NP{:}no' @ \frac{1}{55}, & S/NP{:}plu' @ \frac{1}{55}, & S/NP{:}plu'\,veg' @ \frac{1}{55}, \\
\color{red}{NP{:}veg' @ \frac{9}{55}}, & NP{:}eat' @ \frac{3}{55}, & \color{red}{NP{:}plu'\,veg' @ \frac{2}{55}}, \\
NP{:}plu'\,eat' @ \frac{1}{55}, & NP{:}plu'\,gone' @ \frac{1}{55} & NP{:}plu'\,choc' @ \frac{1}{55}, \\
NP{:}plu'\,no' @ \frac{1}{55}, & NP{:}no' @ \frac{4}{55}, & NP{:}choc' @ \frac{3}{55}, \\
NP{:}gone' @ \frac{1}{55} & & \\
NP\backslash NP{:}plu' @ \frac{3}{55}, & & \\
NP/NP{:}plu' @ \frac{3}{55}, & NP/NP{:}veg' @ \frac{3}{55}, & NP/NP{:}choc' @ \frac{1}{55} \\
\}
\end{array}
$$

Other experiences with approximate but probable meanings

- Planning
- Music
- Vision
- Art

- All high-level cognitive processes are massively serial
- All low-level processes are massively parallel
- Need for symbols seems to be the key (not in Beckett's sense) for the bottleneck (Deacon 1997, 2012)
- Unexpected contribution of grammars in all these domains
- All we need to engender meaning of this sort is a mechanism to execute the grammatical process

Summary

- Humans are doing computations too, when they learn grammar and words
- Searle is a pessimist, and Turing an optimist about artificial systems doing the same thing
- Cognitive science, esp. computational linguistics, shows how the process can be conceived computationally

  : for humans, and for other things with interpretable hardware
- That's their "right stuff"

- A grammar-parser without delivery of meaning is a non-starter
- We can accuse current artificial systems of not doing anything interesting by way of semantics
- That doesn't mean they are incapable.

Sad but true

- There are uncountably many meanings out there
- In "there," we can express countably infinitely many
- Some meanings cannot be expressed
- The kind of meanings that can be expressed cause the same problems for the owners with the right hardware
  - ambiguity in perception and use
  - indeterminacy and likelihood
  - resource boundedness

Concluding remarks

- If we worry about the complexity of a problem, computation as we know today can only give rise to PAC meanings*
    - Only they can be given a causal history of their construal with reasonable resources
    - Valiant (1984): "Inherent algorithmic complexity appears to set serious limits to the range of concepts that can be learned."
    - Transfinite representations can be talked about (e.g. π)
    - but cannot be pinned down (not even the PAC-way)
- Can we look into the brain and see the meaning?
    - Probably not
    - But we might be able to construct a personal history for a meaning associated with a form
    - It is a process

*Probably Approximately Correct. Valiant 1984: "we regard learning as the phenomenon of knowledge acquisition in the absence of explicit programming." The selected hypothesis has high probability for low generalization error.

Thanks to:

- You
- GRAMPLUS project http://groups.inf.ed.ac.uk/gramplus/ (Univ. of Edinburgh, PI: Mark Steedman)
- William Rapaport
- Umut Özge
- Samet Bağçe
- My Advisor, Nicholas V. Findler (1930-2013), who got me into this back in 1987
- Leonard "Aryeh" Faltz, my other mentor, who got me out

Some references

Bozsahin, C. (2006). Grammars, programs, and the Chinese room. In *Proc. of the Int. European Conf. on Computing and Philosophy*, Trondheim, Norway.

Bozsahin, C. (2012). *Combinatory Linguistics*. Berlin/Boston: De Gruyter Mouton.

Deacon, T. W. (1997). *The Symbolic Species: The co-evolution of language and the human brain*. The Penguin Press: London.

Deacon, T. W. (2012). *Incomplete nature: How mind emerged from matter*. Norton & Company.

Landau, B. and L. R. Gleitman (1985). *Language and experience: Evidence from the blind child*. Cambridge, MA: Harvard University Press.

Rapaport, W. J. (1986). Searle's experiments with thought. *Philosophy of science 53*(2), 271–279.

Rapaport, W. J. (2006). How Helen Keller used syntactic semantics to escape from a Chinese Room. *Minds and machines 16*(4), 381–436.

Rey, G. (1986). What's really going on in Searle's "Chinese room". *Philosophical Studies 50*, 169–185.

Rogers, Jr, H. (1959). The present theory of Turing Machine computability. *Journal of the Society for Industrial & Applied Mathematics 7*(1), 114–130.

Searle, J. R. (1980). Minds, brains and programs. *The Behavioral and Brain Sciences 3*, 417–424.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind 59*(236), 433–460.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM 27*(11), 1134–1142.