Complexity and Character of Human Languages

Informatics 2A: Lecture 25

Shay Cohen

19 November 2018

- S[$\lambda x.love(x, Kim)(Sam) \Rightarrow_\beta love(Sam, Kim)$]
  - NP[Sam]
    - NPR[Sam]
      - Sam
  - VP[$\lambda y.\lambda x.love(x, y)(Kim) \Rightarrow_\beta \lambda x.love(x, Kim)$]
    - TV[$\lambda y.\lambda x.love(x, y)$]
      - loves
    - NP[Kim]
      - NPR[Kim]
        - Kim

Last Class: Adding Determiners to Semantics

Grammar III

S $\rightarrow$ NP VP	{ NP.Sem(VP.Sem) }	t
VP $\rightarrow$ TV NP	{ TV.Sem(NP.Sem) }	$< e, t >$
NP $\rightarrow$ Sam	{ $\lambda P.P(Sam)$ }	$<< e, t >, t >$
NP $\rightarrow$ Det Nom	{ Det.Sem(Nom.Sem) }	$<< e, t >, t >$
Det $\rightarrow$ a	{ $\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x)$ }	$<< e, t >, << e, t >, t >>>$
Det $\rightarrow$ every	{ $\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x)$ }	$<< e, t >, << e, t >, t >>>$
Nom $\rightarrow$ N	{ N.Sem }	$< e, t >$
Nom $\rightarrow$ A Nom	{ $\lambda x.$ Nom.Sem(x) & A.Sem(x) }	$< e, t >$
TV $\rightarrow$ loves	{ $\{\lambda R.\lambda z.\ R(\lambda w.\ loves(z, w))\}$	$<<< e, t >, t >, < e, t >>$
N $\rightarrow$ woman	{ $\lambda z.woman(z)$ }	$< e, t >$
A $\rightarrow$ tall	{ $\lambda z.tall(z)$ }	$< e, t >$

Can add similar entries for ‘student’, ‘computer’, ‘has access to’.

Example

The semantics for ‘every student has access to a computer’.

Example

The semantics for 'every student has access to a computer'.

$$\begin{array}{ll} \text{every student} & (\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x)) \\ & \rightarrow_\beta \quad \lambda P.\forall x.\, student(x) \Rightarrow P(x) \end{array}$$

Example

The semantics for 'every student has access to a computer'.

every student $(\lambda Q.\lambda P.\,\forall x.Q(x) \Rightarrow P(x))(\lambda x.\mathit{student}(x))$

$\rightarrow_\beta \quad \lambda P.\,\forall x.\,\mathit{student}(x) \Rightarrow P(x)$

a computer $(\lambda Q.\lambda P.\,\exists x.Q(x) \wedge P(x))(\lambda x.\mathit{computer}(x))$

$\rightarrow_\beta \quad \lambda P.\,\exists x.\,\mathit{computer}(x) \wedge P(x)$

Example

The semantics for ‘every student has access to a computer’.

every student $(\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x))$
$\rightarrow_\beta \quad \lambda P.\forall x.\, student(x) \Rightarrow P(x)$

a computer $(\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x))(\lambda x.computer(x))$
$\rightarrow_\beta \quad \lambda P.\exists x.\, computer(x) \wedge P(x)$

h.a.t. a computer $\cdots \rightarrow_\beta \cdots$
$\rightarrow_\beta \lambda z.\exists x.\, computer(x) \wedge h\_a\_t(z, x)$

Example

The semantics for ‘every student has access to a computer’.

every student $(\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x))$
$\rightarrow_\beta \quad \lambda P.\forall x.\, student(x) \Rightarrow P(x)$

a computer $(\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x))(\lambda x.computer(x))$
$\rightarrow_\beta \quad \lambda P.\exists x.\, computer(x) \wedge P(x)$

h.a.t. a computer $\cdots \rightarrow_\beta \cdots$
$\rightarrow_\beta \lambda z.\exists x.\, computer(x) \wedge h\_a\_t(z, x)$

(whole sentence) $\cdots \rightarrow_\beta \cdots$
$\rightarrow_\beta \forall x.\, student(x) \Rightarrow \exists y.\, computer(y) \wedge h\_a\_t(x, y)$

Example

The semantics for 'every student has access to a computer'.

every student $(\lambda Q.\lambda P.\,\forall x.\,Q(x) \Rightarrow P(x))(\lambda x.student(x))$
$\rightarrow_\beta \quad \lambda P.\,\forall x.\,student(x) \Rightarrow P(x)$

a computer $(\lambda Q.\lambda P.\,\exists x.\,Q(x) \land P(x))(\lambda x.computer(x))$
$\rightarrow_\beta \quad \lambda P.\,\exists x.\,computer(x) \land P(x)$

h.a.t. a computer $\cdots \quad \rightarrow_\beta \quad \cdots$
$\rightarrow_\beta \lambda z.\exists x.\,computer(x) \land h\_a\_t(z, x)$

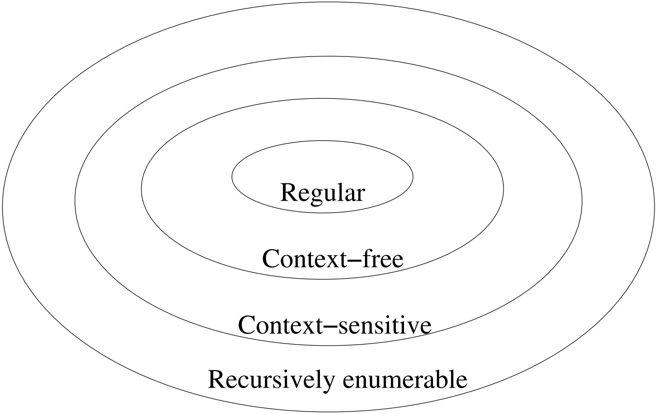
(whole sentence) $\cdots \quad \rightarrow_\beta \quad \cdots$
$\rightarrow_\beta \forall x.\,student(x) \Rightarrow \exists y.\,computer(y) \land h\_a\_t(x, y)$

Note: In the last β-step, we've renamed 'x' to 'y' to avoid capture.

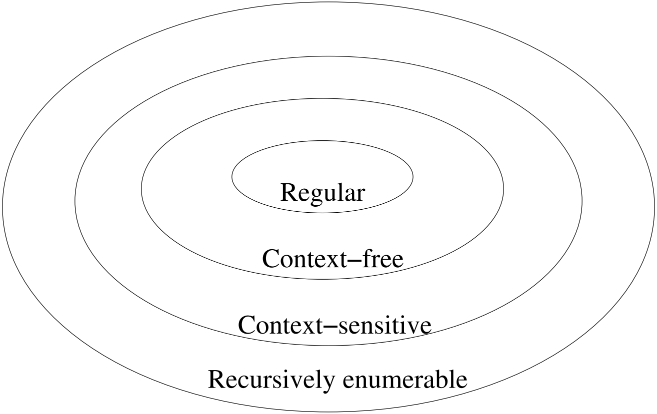
Semantics: Things to Think About

- Why is it $\lambda y, x.\mathrm{loves}(x, y)$ and not $\lambda x, y.\mathrm{loves}(x, y)$?
- Given two types of the three types (for a parent node and two children nodes), how do we determine the third one?
- How did we determine the type of “every” through type raising? What does Q and P mean in its definition?
- How can we change variable names to make the handling of scope easier when using β-reduction?

Recap: The Chomsky hierarchy



Recap: The Chomsky hierarchy



Where exactly do human languages fit within this complexity hierarchy?

Infinity and languages

How 'complex' are human languages?

The potential infiniteness of language has been recognized for centuries (by Galileo, Descartes, von Humboldt...)

There is no longest sentence!

Mary thinks that John thinks that George thinks that Mary thinks that this course is boring!
I woke up and had a coffee and got dressed and checked facebook and walked in the park and ate lunch . . .

Is Natural Language Regular?

Of course, many infinite languages are regular, e.g. $\{a^n | n \geq 0\}$ is regular. But what about natural languages?

E.g. Is English a regular language?

Is Natural Language Regular?

Of course, many infinite languages are regular, e.g. $\{a^n | n \geq 0\}$ is regular. But what about natural languages?

E.g. Is English a regular language?

Challenge: How can we even answer the question, given that we don't have a complete mathematical 'definition' of English? (And anyway, English is 'fuzzy at the edges'.)

Fortunately, we don't need one. Just need to agree that certain sentences are definitely in, and certain others are definitely out.

We can then show that no regular language includes all the former, but excludes all the latter.

Is Natural Language Regular?

Of course, many infinite languages are regular, e.g. $\{a^n | n \geq 0\}$ is regular. But what about natural languages?

E.g. Is English a regular language?

Challenge: How can we even answer the question, given that we don't have a complete mathematical 'definition' of English? (And anyway, English is 'fuzzy at the edges'.)

Fortunately, we don't need one. Just need to agree that certain sentences are definitely in, and certain others are definitely out.

We can then show that no regular language includes all the former, but excludes all the latter.

Tools:

- Pumping Lemma
- Intersection property: If L and L' are regular then so is $L \cap L'$. (Hence if L is regular but $L \cap$ English isn't regular, then English can't be regular.)

Is English Regular?

Centre-embedding

[The cat_1 likes tuna fish_1].
[The cat_1 [the dog_2 chased_2] likes tuna fish_1].
[The cat_1 [the dog_2 [the rat_3 bit_3] chased_2] likes tuna fish_1].

Is English Regular?

Centre-embedding

[The cat_1 likes tuna fish_1].
[The cat_1 [the dog_2 chased_2] likes tuna fish_1].
[The cat_1 [the dog_2 [the rat_3 bit_3] chased_2] likes tuna fish_1].

Consider $L = \{(\text{the N})^n \ \text{TV}^m \ \text{likes tuna fish} \mid n, m \geq 0\}$

where N = { cat, dog, rat, elephant, kangaroo ... }
TV = { chased, bit, admired, ate, befriended ... }

Clearly L is regular. However, $L \cap$ English is the language

$$\{(\text{the N})^n \ \text{TV}^{n-1} \ \text{likes tuna fish} \mid n \geq 1\}$$

Can use pumping lemma to show L is not regular.

Assumption 1. "(the N)^n TV^m likes tuna fish" is ungrammatical for $m \neq n - 1$.
Assumption 2. "(the N)^n TV^{n-1} likes tuna fish" is grammatical for all $n \geq 1$. (Is this reasonable? You decide!)

Are natural languages context-free?

Are context-free grammars sufficient for modelling NL grammar? Or are there aspects of NLs that they can't capture?

How would we know if there were such aspects? Again, there are tools for showing a language isn't context-free:

- Context-free pumping lemma (Lecture 29). Using this, we can show (for example) that

$$\{a^n b^m c^n d^m \mid n, m \geq 0\}$$

  is **not** context-free.
- Intersection property: If L is regular and L' is context-free, then $L \cap L'$ is context-free.
  (Idea: can 'combine' an NPDA for L' with an NFA for L to get an NPDA for $L \cap L'$.)

Note in passing that the intersection of two context-free languages **needn't** be context-free. (Above trick doesn't work: only allowed one stack!)

Non-context-freeness in natural languages

Non-context-freeness in natural languages

In Swiss German, some verbs (e.g. *let*, *paint*) take an object in accusative form, while others (e.g. *help*) take an object in dative form. The nouns are case-marked even in subordinate clauses, which in Swiss-German, can exhibit cross-serial dependencies.

Cross-serial dependencies

. . . das mer	d'chind	em Hans	es huus	lönd	hälfe	aastriiche
. . . that we	the children	Hans	the house	let	help	paint
	NP-ACC	NP-DAT	NP-ACC	V-ACC	V-DAT	V-ACC

. . . that we let the children help Hans paint the house

Back to question: Is Natural Language Context Free?

Claim 1. Swiss German subordinate clauses can have a structure in which all the Vs follow all the NPs.

Back to question: Is Natural Language Context Free?

Claim 1. Swiss German subordinate clauses can have a structure in which all the Vs follow all the NPs.

Claim 2. Among such sentences, those with all dative NPs preceding all accusative NPs, and all dative-subcategorizing Vs preceding all accusative-subcategorizing Vs are acceptable.

Back to question: Is Natural Language Context Free?

Claim 1. Swiss German subordinate clauses can have a structure in which all the Vs follow all the NPs.

Claim 2. Among such sentences, those with all dative NPs preceding all accusative NPs, and all dative-subcategorizing Vs preceding all accusative-subcategorizing Vs are acceptable.

Claim 3. The number of Vs requiring dative objects must equal the number of dative NPs and similarly for accusatives.

Back to question: Is Natural Language Context Free?

Claim 1. Swiss German subordinate clauses can have a structure in which all the Vs follow all the NPs.

Claim 2. Among such sentences, those with all dative NPs preceding all accusative NPs, and all dative-subcategorizing Vs preceding all accusative-subcategorizing Vs are acceptable.

Claim 3. The number of Vs requiring dative objects must equal the number of dative NPs and similarly for accusatives.

Claim 4. An arbitrary number of Vs can occur in a subordinate clause. (cf. similar claim in our proof of English context-freeness)

Back to question: Is Natural Language Context Free?

Claim. Swiss-German is not context-free.

Back to question: Is Natural Language Context Free?

Claim. Swiss-German is not context-free.

Sketch of proof. Represent dative NPs, accusative NPs, dative-subcategorizing Vs, and accusative-subcategorizing Vs by symbols A, B, C, and D, respectively.
Then among all constructions of the form $A^* B^* C^* D^*$, the grammatically acceptable ones are exactly those of the form $A^n B^m C^n D^m$.
So intersecting Swiss German with a suitable regular language yields the set of strings $A^n B^m C^n D^m$.

But this language is known not to be context-free. Since context-free languages are closed under intersection with regular languages, Swiss-German can't be context-free either!

Review

Chomsky Hierarchy: classifies languages on scale of complexity:

- Regular languages: those whose phrases can be 'recognized' by a finite state machine.
- Context-free languages: those describable via 'context-free rules' $X \to \beta$, where $X \in N$ and $\beta \in (N \cup \Sigma)^*$.
  Many aspects of PLs and NLs can be described at this level;
- Context-sensitive languages: those describable via 'context-sensitive rules' $\alpha X \gamma \to \alpha \beta \gamma$.
  More than enough for all known features of FLs and NLs. (E.g. typing/scoping rules in PLs; Swiss-German crossing dependencies.)
- Recursively enumerable languages: *all* languages that can in principle be defined via mechanical rules.

Strong and Weak Adequacy

Questions about the formal complexity of language are about the computational power of syntax, as represented by a grammar that's adequate for it.

A strongly adequate grammar

- generates all and only the strings of the language;
- assigns them the "right" structures — e.g. ones that allow us to compute a correct representation of meaning (as in previous lecture).

A weakly adequate grammar

generates all and only the strings of a language but doesn't necessarily give a correct (insightful) account of their structures.

Weaker examples

Swiss-German 'crossing dependencies' are non-context-free in a very strong sense: no CFG is even weakly adequate for modelling them.

There are other phenomena that in theory *could* be modelled using CFGs, though it seems unnatural to do so. E.g. a versus an in English.

English.	a banana	an apple
	a large apple	an exceptionally large banana

Over-simplifying a bit: a before consonants, an before vowels.

In theory, we could use a context-free grammar:

$$\begin{array}{rcl} \mathrm{NP} & \rightarrow & \textbf{a}\ \mathrm{NP1}^c \\ \mathrm{NP1}^c & \rightarrow & \mathrm{N}^c \mid \mathrm{AP}^c\ \mathrm{NP1} \\ \mathrm{AP}^c & \rightarrow & \mathrm{A}^c \mid \mathrm{Adv}^c\ \mathrm{AP} \end{array} \qquad \begin{array}{rcl} \mathrm{NP} & \rightarrow & \textbf{an}\ \mathrm{NP1}^v \\ \mathrm{NP1}^v & \rightarrow & \mathrm{N}^v \mid \mathrm{AP}^v\ \mathrm{NP1} \\ \mathrm{AP}^v & \rightarrow & \mathrm{A}^v \mid \mathrm{Adv}^v\ \mathrm{AP} \end{array}$$

But more natural to use context-sensitive rules, e.g.

$$\begin{array}{rcl} \mathrm{DET}\ [\text{c-word}] & \rightarrow & \textbf{a}\ [\text{c-word}] \\ \mathrm{DET}\ [\text{v-word}] & \rightarrow & \textbf{an}\ [\text{v-word}] \end{array}$$

Linear indexed grammars (LIGs) are a formalism more powerful than CFGs, but much less powerful than an arbitrary CSGs. Think of them as mildly context sensitive grammars. These seem to suffice for NL phenomena.

Definition

An indexed grammar has three disjoint sets of symbols: terminals, non-terminals and indices.

An index is a stack of symbols that can be passed from the LHS of a rule to its RHS, allowing counting and recording what rules were applied in what order. So think of LIGs as CFGs where a little bit of ‘context information’ may be passed down to subphrases.

Summary

- We can argue quite rigorously about the complexity of NLs, even without having a complete 'definition' of any NL.
- NLs make frequent use of nested structures, which can be used to show they can't be regular.
- Some NLs contain constructs which (in a strong sense) surpass the power of context-free grammars.
- Many NLs contain features that could in theory be modelled by CFGs, but are in practice better treated in some other way.
- NLs appear to surpass the power of context-free languages, but only just. E.g. the mild form of context-sensitivity captured by LIGs seems at least weakly adequate for NL structures.