Languages

Languages: syntax, semantics, pragmatics

- **Pragmatics:** new_var = map(lambda x: x - 2, [4, 5, 6])
- **Semantics**:
    - This is a valid sentence in English.
    - The worst part and clumsy looking for whoever heard light.
    - Twas brillig, and the slithy toves did gyre and gimble in the wabe.
    - Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor …
- **Grammar (syntax)**:
    - I can has cheezburger?
    - I nevr mkae tipos and erors in my sentencs.
    - I’m chuffed to bits seeing you! Do ya wanna watch some telly together, bro?
    - I’ll txt w/my ETA 2U.

Definitions

syntax guards word order

[formal] **grammar** - describes how to form strings from a language's *alphabet* that are valid according to the language's *syntax*. = set of **rules**, way to express syntax

formal language - set of all strings *allowed* by a grammar. = **satisfy** rules

Grammar is a <**Σ** - terminals, **N** - nonterminals, **P** - productions, **S** - start symbol>

Sample language

Σ: {0 … 9}

N: {NZ, Z}

P:

- NZ => NZ + NZ
- NZ => NZ + Z
- NZ => 1...9
- Z => 0
- S => NZ
- S => Z

S = {Natural numbers}

FSA - finite state machine / automaton

$(\Sigma, S, s_0, \delta, F)$

Σ — input alphabet (commands you can recieve)

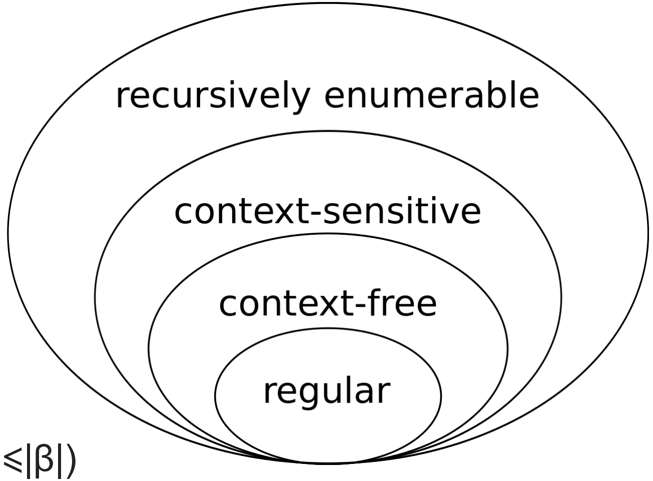
S — states

s_0 — initial state

δ — transition function $\delta : S \times \Sigma \rightarrow S$ (compare to *productions*)

F ⊂ S — set of final states

Chomsky Normal Form (CNF) and Chomsky hierarchy



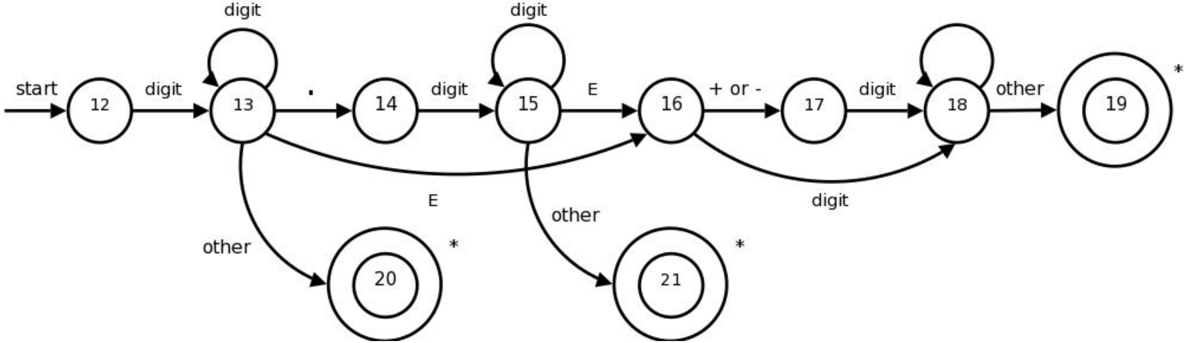
0. **Recursively enumerable** (any types of productions)

1. **Context-sensitive** αAβ → αγβ

Also noncontracting grammar (α→β, где α,β∈{Σ∪N}+ и |α|⩽|β|)

2. **Context-free** $A \to \alpha$

3. **Regular** A → a or A → aB | A → Ba (exceptionally)



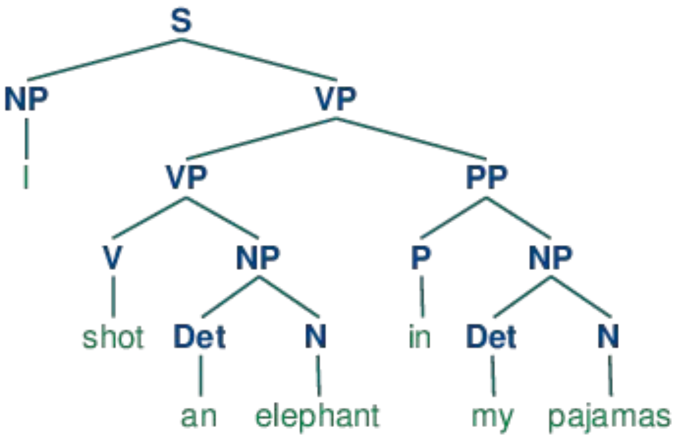
Extended Backus-Naur Form (EBNF)

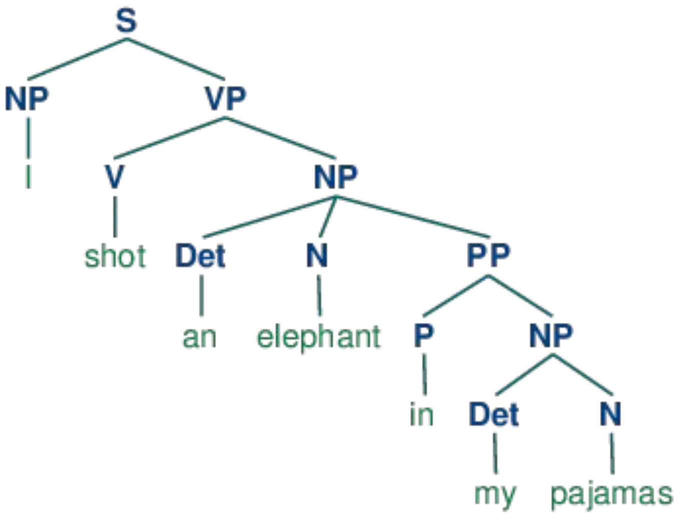
```
A = B,C.          # concat

A = B|C|D.        # one of

A = [B].          # 0/1

A = {B}.          # 0+

A = B{B}.         # 1+

A = (B|C)(D|E).   # grouping (to avoid service NT)
```

Context-free grammar

```
S -> NP VP
PP -> P NP
NP -> Det N | Det N PP | 'I'
VP -> V NP | VP PP
Det -> 'an' | 'my'
N -> 'elephant' | 'pajamas'
V -> 'shot'
P -> 'in'
```

Syntax tree


S
NP
VP
I
VP
PP
V
NP
P
NP
shot
Det
N
in
Det
N
an
elephant
my
pajamas


S
NP
VP
I
V
NP
shot
Det
N
PP
an
elephant
P
NP
in
Det
N
my
pajamas