Introduction to Information Sciences

Introduction to the Course

M. Cuturi, X. Liang

Summary of the Course

- Provide a wide overview of
  - natural language processing & linguistics
  - information theory
  - computer science
  - artificial intelligence
  - pattern recognition, speech processing $etc.$
- This course is designed to introduce the activities of the *Intelligence Science and Technology* department to the rest of the university.

Grading

- Two lecturers $\rightarrow$ overall grade split into two respective parts
- each part is equally weighted.

- Attendance is important.
  - Total of 13 classes.
  - You should attend at least ≥ 10 classes. you have 3 jokers.
  - Below that, -10% per missed class.
  - Below ≤ 7, no credit.
- Grading will be based on reports. 2 reports in my course.

Before we proceed...

Questions?...

Let's start with a little Quizz

Question 1

Consider the following algorithm

- $x \leftarrow 0$;
- **For** $i = 1, \cdots, 3$
  - $x \leftarrow x + i$;
- **endFor**

What is the value of x once this algorithm has been executed?

1. 3 **2.** 0 **3.** 6 **4.** 1 **5.** 12

Let's start with a little Quizz

Question 1

Consider the following algorithm

- $x \leftarrow 0$;
- **For** $i = 1, \cdots, 3$
  - $x \leftarrow x + i$;
- **endFor**

What is the value of x once this algorithm has been executed?

1. 3 **2.** 0 **3.** **6** **4.** 1 **5.** 12

Let's start with a little Quizz

Question 2

Consider the following algorithm

- $x \leftarrow 2$;
- **For** $i = 1, \cdots, 3$
  - **If** $i \leq x$,
    - $x \leftarrow x - i$;
  - **Else**
    - $x \leftarrow x + i$;
  - **endIf**
- **endFor**

What is the value of x once this algorithm has been executed?

1. 2 **2.** 0 **3.** 4 **4.** -1 **5.** 3

Let's start with a little Quizz

Question 2

Consider the following algorithm

- $x \leftarrow 2$;
- **For** $i = 1, \cdots, 3$
  - **If** $i \leq x$,
    - $x \leftarrow x - i$;
  - **Else**
    - $x \leftarrow x + i$;
  - **endIf**
- **endFor**

What is the value of x once this algorithm has been executed?

1. 2 **2.** **0** **3.** 4 **4.** -1 **5.** 3

Introduction to Information Sciences

Natural Language Processing
Formal Languages

mcuturi@i.kyoto-u.ac.jp

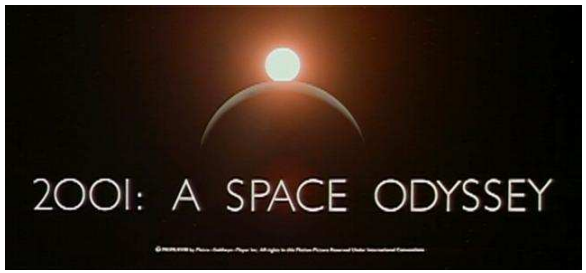
Summary

- Illustrate the difficulties tackled by computational linguistics
  - Define a few of the problems commonly studied
- Introduce formal language theory & Automata
  - formal languages
  - formal grammars
  - Chomsky hierarchy

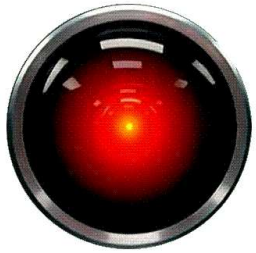
Sources for these slides: A. McCallum's (UMass) online lectures, Wikipedia, Jurafsky/Martin

We start with an example: HAL

- An example taken from a famous movie and book:



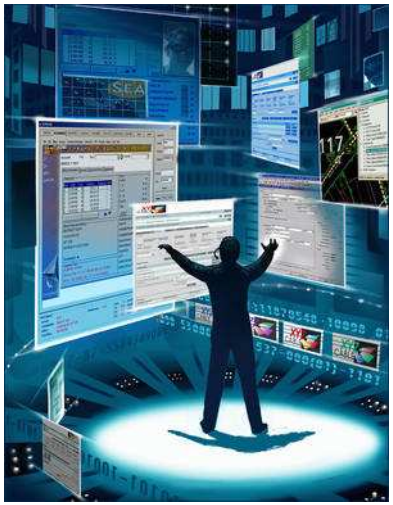
- Let's check a few scenes:



2001 was shot in 1968

A few years after 2001, what sounds familiar, if not outdated about HAL?

- Graphic capabilities?.. We have much better. The future rather looks like this...



- Chess? 2006, Deep Fritz and before, late 90's, Deep Blue

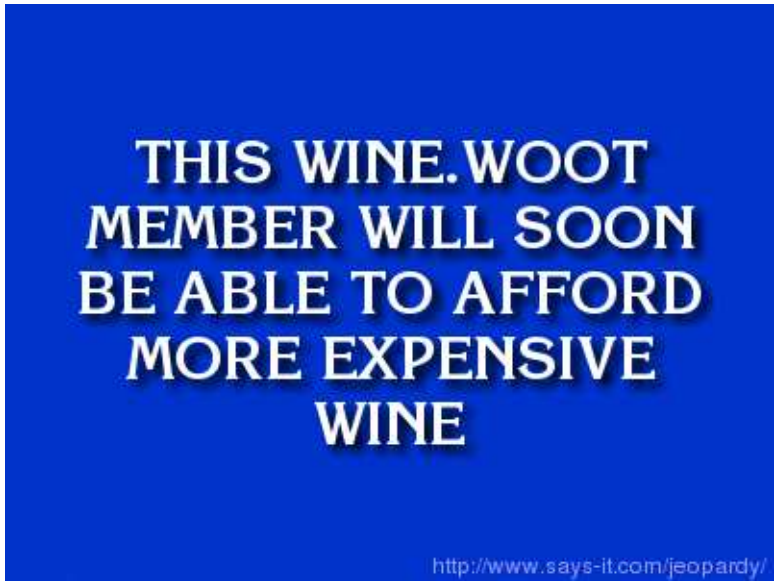
2001 was shot in 1968

- What still sounds difficult to achieve is HAL's **articulated syntax**...

David Bowman:
Open the pod bay doors, Hal.
HAL:
I'm sorry, Dave, Im afraid I cant do that.
David Bowman:
What are you talking about, Hal?
...HAL:
I know that you and Frank were planning to disconnect me,
and I'm afraid that's something I cannot allow to happen.

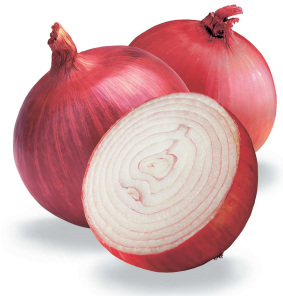
- The machine is also displaying **intelligence**. See Turing's test.
- Yet, **why does language seem more difficult to reach than chess?**

Recent Progress



Layers of Computational Linguistics

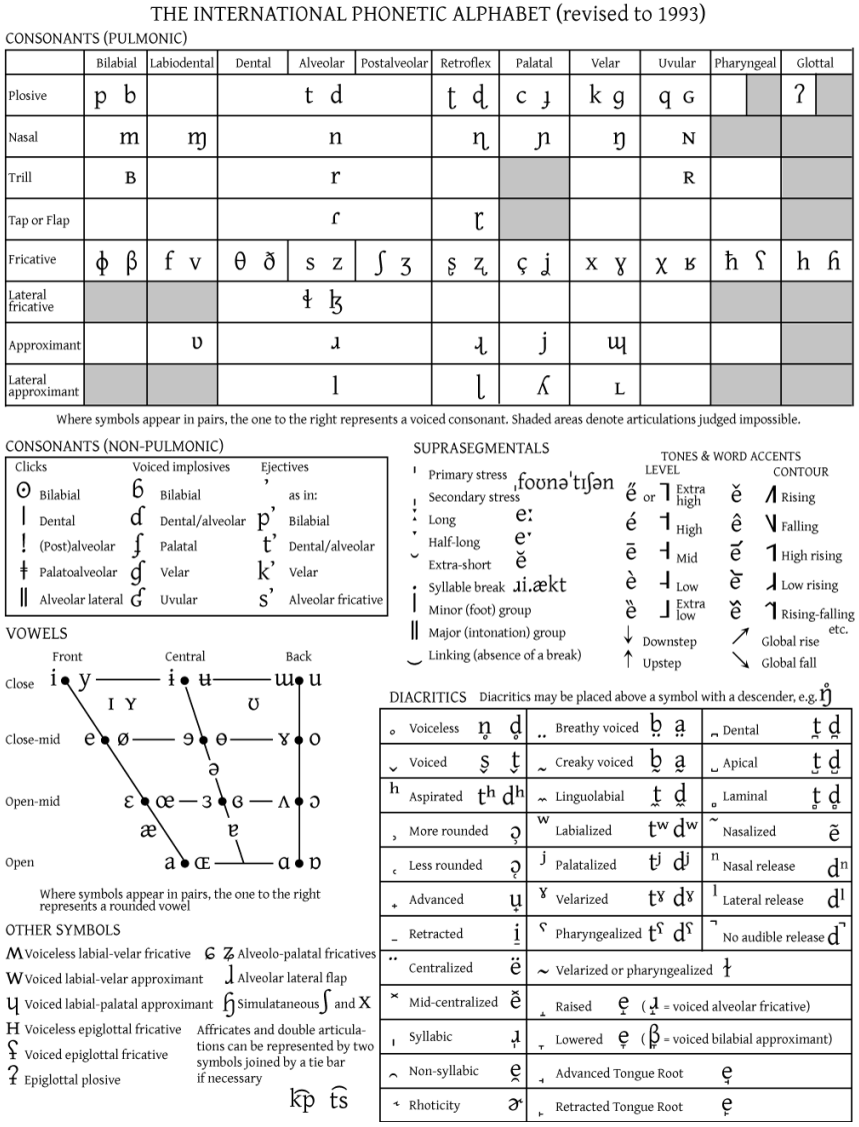
Complex and multilayered system, each layer a different study field



- Phonetics
- Phonology
- Morphology
- Syntax
- Semantics
- Pragmatics
- Discourse

Phonetics

Study of language sounds, physical aspects.



THE INTERNATIONAL PHONETIC ALPHABET (revised to 1993)

CONSONANTS (PULMONIC)

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap				ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks		Voiced implosives		Ejectives	
ʘ	Bilabial	ɓ	Bilabial	’	as in:
ǀ	Dental	ɗ	Dental/alveolar	pʼ	Bilabial
ǃ	(Post)alveolar	ʄ	Palatal	tʼ	Dental/alveolar
ǂ	Palatoalveolar	ɠ	Velar	kʼ	Velar
ǁ	Alveolar lateral	ʛ	Uvular	sʼ	Alveolar fricative

SUPRASEGMENTALS

ˈ	Primary stress	ˌfoʊnəˈtɪʃən	
ˌ	Secondary stress		
ː	Long	eː	
ˑ	Half-long	eˑ	
̆	Extra-short	ĕ	
.	Syllable break	ɹi.ækt	
|	Minor (foot) group		
‖	Major (intonation) group		
‿	Linking (absence of a break)		

TONES & WORD ACCENTS

LEVEL		CONTOUR	
e̋ or ˥	Extra high	ě	˩˥ Rising
é ˦	High	ê	˥˩ Falling
ē ˧	Mid	e᷄	˥˧ High rising
è ˨	Low	e᷅	˧˩ Low rising
ȅ ˩	Extra low	e᷈	˥˧˥ Rising-falling
↓	Downstep	↗	Global rise
↑	Upstep	↘	Global fall

etc.

VOWELS

	Front	Central	Back
Close	i y	ɨ ʉ	ɯ u
	ɪ ʏ	ʊ	
Close-mid	e ø	ɘ ɵ	ɤ o
		ə	
Open-mid	ɛ œ	ɜ ɞ	ʌ ɔ
	æ	ɐ	
Open	a œ	ɑ	ɒ

Where symbols appear in pairs, the one to the right represents a rounded vowel

OTHER SYMBOLS

ʍ	Voiceless labial-velar fricative
w	Voiced labial-velar approximant
ɥ	Voiced labial-palatal approximant
ʜ	Voiceless epiglottal fricative
ʢ	Voiced epiglottal fricative
ʡ	Epiglottal plosive
ɕ ʑ	Alveolo-palatal fricatives
ɺ	Alveolar lateral flap
ɧ	Simulataneous ʃ and x

Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary

k͡p t͡s

DIACRITICS Diacritics may be placed above a symbol with a descender, e.g. ŋ̊

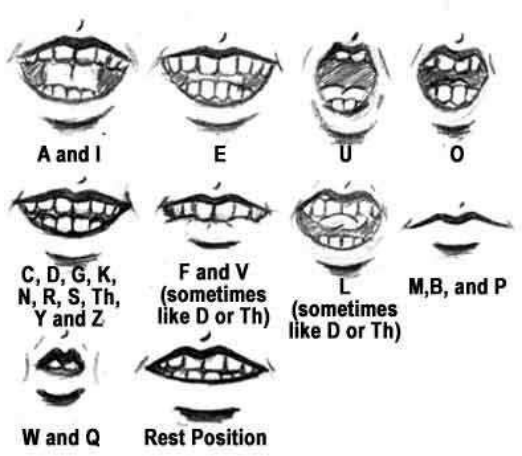
̥	Voiceless	n̥ d̥	̤	Breathy voiced	b̤ a̤
̬	Voiced	s̬ t̬	̰	Creaky voiced	b̰ a̰
ʰ	Aspirated	tʰ dʰ	̼	Linguolabial	t̼ d̼
̹	More rounded	ɔ̹	ʷ	Labialized	tʷ dʷ
̜	Less rounded	ɔ̜	ʲ	Palatalized	tʲ dʲ
̟	Advanced	u̟	ˠ	Velarized	tˠ dˠ
̠	Retracted	i̠	ˤ	Pharyngealized	tˤ dˤ
̈	Centralized	ë	~	Velarized or pharyngealized	ɫ
̽	Mid-centralized	e̽	̝	Raised	e̝ (ɹ̝ = voiced alveolar fricative)
̩	Syllabic	n̩	̞	Lowered	e̞ (β̞ = voiced bilabial approximant)
̯	Non-syllabic	e̯	̘	Advanced Tongue Root	e̘
˞	Rhoticity	ɚ	̙	Retracted Tongue Root	e̙

̪	Dental	t̪ d̪
̺	Apical	t̺ d̺
̻	Laminal	t̻ d̻
̃	Nasalized	ẽ
ⁿ	Nasal release	dⁿ
ˡ	Lateral release	dˡ
̚	No audible release	d̚

Phonology

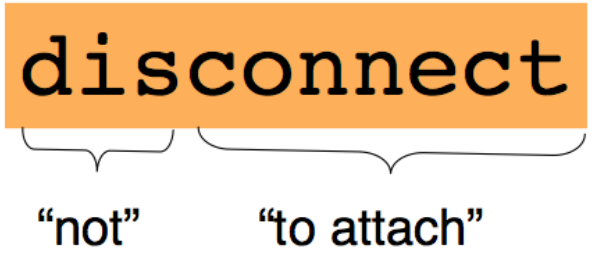
Study of sound **structure** of language.

- Identify units of sounds, in **different** human languages.
  - **phonemes**,
  - **syllables**,
- Phonemes are a major difference between animal language and human language.
- Useful for instance in animations. Phonemes in english:



Morphology

Study of morphemes, the minimal units of linguistic form and meaning



- Important for compounded languages $e.g.$ Turkish:

uygarlastiramadiklarimizdanmissinizcasina

uygar las tir ama dik lar imiz dan mis siniz casina

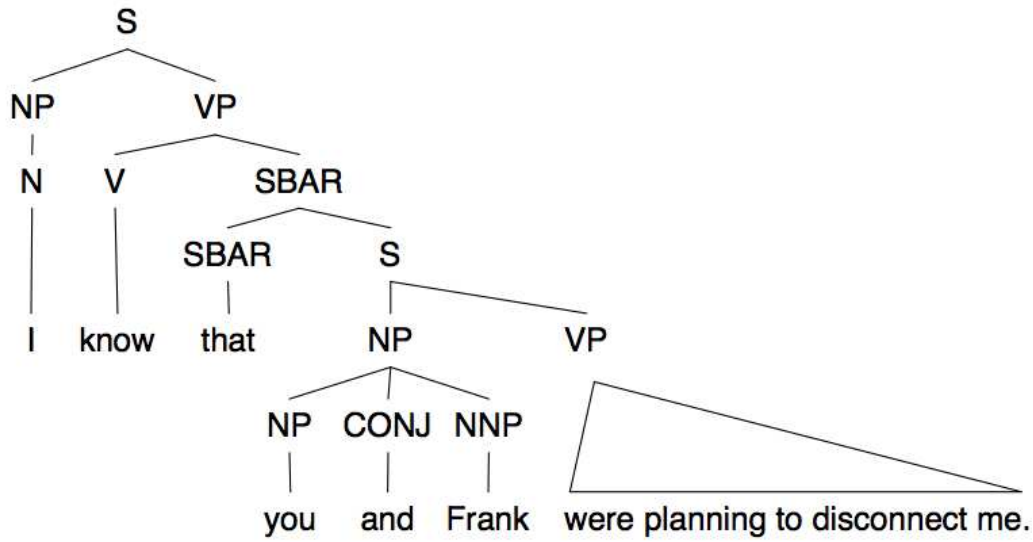
(behaving) as if you are among those whom we could not civilize

- In chinese, chinese characters = morphems = basic semantic unit

Syntax

- From words to sentences:

I know that you and Frank were planning to disconnect me.



- Of course, the structure (the syntax) of the following sentence is also correct

You know me–Frank and I were planning to disconnect that.

Semantics

Study of meaning, the minimal units of linguistic form and meaning

- The meaning of

  I know that you and Frank were planning to disconnect me.

  can be summarized as

  - an *action*, **disconnect**,
  - performed by an *actor*, **you and Franck**,
  - on an *object*, **me**

- In computer science, different syntaxes for the same operation:
  `x += y` (C, Java, Perl, Python, Ruby, etc.)
  `x := x + y` (Pascal)
  `LET X = X + Y` (early BASIC)
  `x = x + y` (MATLAB, most BASIC dialects, Fortran)
  `(incf x y)` (Common Lisp)

Pragmatics

The study of how language is used to accomplish goals within a given **context**

- What is the practical outcome of a sentence as

  Im sorry, Dave, Im afraid I cant do that.

  given the contex?

- The sentence "You have a green light" can have different meanings:
  - It could mean you are holding a green light bulb.
  - Or that you have a green light to drive your car.
  - Or it could be indicating that you can go ahead with the project.
  - Or that your body has a green glow

Discourse

Study of linguistic units which are larger than single **utterances**

- Capture the different turns, threads, changes in the conversation

David Bowman:
Open the pod bay doors, Hal.
HAL:
Im sorry, Dave, Im afraid I cant do that.
David Bowman:
What are you talking about, Hal?
...HAL:
I know that you and Frank were planning to disconnect me, and I'm afraid that's something I cannot allow to happen.

Languages contain all possible utterances

- Here are sentences in the english language,
  - The man took the book.
  - This sentence is not true.
  - The horse was galloping in the prairie
- Here are sentences which are **not** part of it
  - The true the eat lot looks bird.
  - sense any make not does sentence this
  - dfdlkfh lkjer leREQ ARlkjdf
- A few different kinds of language:
  - Natural languages language that arises in an **unpremeditated** manner as the product of the human innate facility to communicate. Can be spoken, signed, written etc..
  - Constructed languages constructed languages as auxiliary languages such as **esperanto** or artistic languages ($e.g.$ in fiction)
  - Formal languages: languages that computers can parse and understand.
- The latter is the family of languages we will study the most in these 2 lectures.

Seen from a computer, a language is a set

- We start with the formal idea of alphabets, a set of tokens

$$\Sigma = \{a, b, c, d, e, f, g, \cdots, z, , \cdots\} \text{ or,}$$
$$\Sigma = \{0, 1\} \text{ or,}$$
$$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, \ln, \exp, \cdots\}.$$

- and use the Kleene operator as a shortcut for

$$\Sigma^\star = \{x \in \Sigma^n, n \in \mathbf{N}\}.$$

- A formal language L is a **subset** of $\Sigma^\star$.

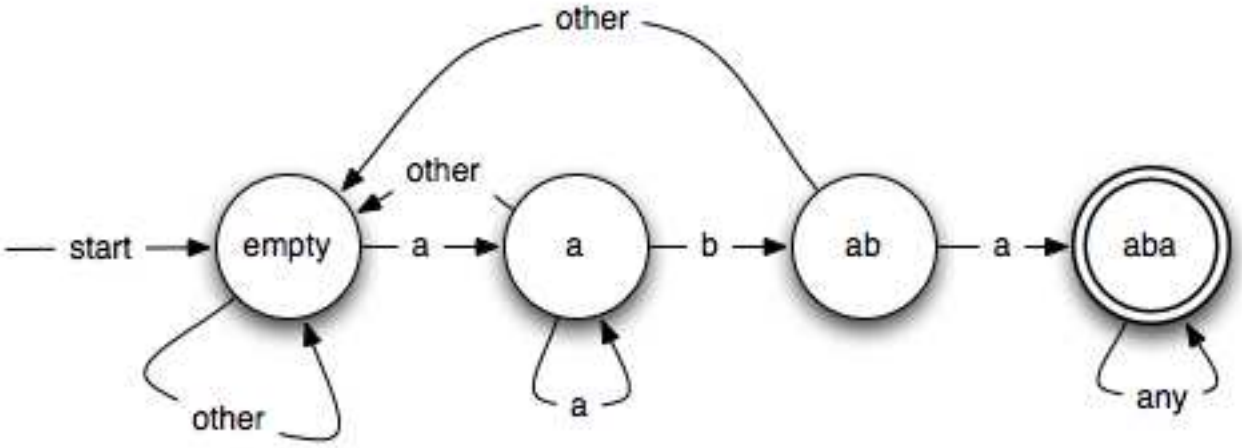
Example of a language

Rules can describe a formal language L

- Consider the language L defined as
  - The alphabet $= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, =$:
  - Every nonempty string that does not contain $+$ or $=$ and does not start with 0 is in L.
  - The string 0 is in L.
  - A string containing $=$ is in L if and only if there is exactly one $=$, and it separates two valid strings in L.
  - A string containing $+$ but not $=$ is in L if and only if every $+$ in the string separates two valid strings in L.
  - No string is in L other than those implied by the previous rules.
- With such rules,
  - "23+4=555" is in L,
  - "d433+2e2" is not,
  - "=234=+" is not.
- **no meaning** yet though. Only notion of belonging or not to a language.

Formal languages = typology of such rules

- Other ways to define a language from an alphabet:
- For instance, a language can be given as
  - all strings generated by a **formal grammar**;
  - all strings accepted by some **automaton**, in the example the automaton can generate the language of all words containing at least "*aba*" once



  - all strings described or matched by a particular **regular expression**;
  - all strings for which some decision procedure (an algorithm that asks a sequence of related YES/NO questions) produces the answer YES.

Typical questions asked about such formalisms

- What is their expressive power? (Can formalism X describe every language that formalism Y can describe? Can it describe other languages?)
- What is their recognizability? (How difficult is it to decide whether a given word belongs to a language described by formalism X?)
- What is their comparability? (How difficult is it to decide whether two languages, one described in formalism X and one in formalism Y, or in X again, are actually the same language?).

Formal grammar

A **formal grammar** is a set of rules which generate **formal languages**, defined by:

- a finite set of terminal symbols,
- a finite set of nonterminal symbols,
- a start symbol which is a nonterminal symbol,
- a finite set of production rules:

$$\text{Rule}: \quad \cdots \rightarrow \cdots$$

where the dots are arbitrary symbols.

Formal grammar

- How?
  - Start with the start symbol.
  - Apply any rule by replacing an occurrence of the symbols on the left-hand side of the rule with those that appear on the right-hand side.
- A sequence of rule applications is called a derivation.

> Such a grammar defines the formal language:
> **all words** consisting **solely of terminal symbols**
> which can be reached by **a derivation**
> from the **start symbol**.

- Usually, NONTERMINALS are represented by uppercase letters,
- terminals by lowercase letters,
- the start symbol by S.

Formal grammar Example

- For example, the grammar with
  - terminals $\{a, b\}$,
  - nonterminals $\{S, A, B\}$, starting S,
  - production rules
    - $S \to ABS$
    - $S \to \varepsilon$ (where ε is the empty string)
    - $BA \to AB$
    - $BS \to b$
    - $Bb \to bb$
    - $Ab \to ab$
    - $Aa \to aa$

  defines the language of all words of the form $a^n b^n$.
- simpler grammar that defines the same language:
  - Terminals $\{a, b\}$,
  - Nonterminals $\{S\}$, Start symbol S, Production rules
    - $S \to aSb$
    - $S\varepsilon$

Chomsky Hierarchy of Formal Languages

- Type-0 : all grammars.
- Type-1: $\alpha A \beta \rightarrow \alpha \gamma \beta$ where γ cannot be empty. $S \rightarrow \varepsilon$ is allowed iff S does not appear on the right side of a rule.
- Type-2 $A \rightarrow \gamma$ where γ a string of terminals and nonterminals.
- Type-3: Nonterminals can only appear on one side, $S \rightarrow \varepsilon$ is allowed iff S does not appear on the right side of a rule.

Grammar	Languages	Automaton	Production rules (constraints)
Type-0	Recursively enumerable	Turing machine	$\alpha \rightarrow \beta$(no restrictions)
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Type-2	**Context-free**	Non-deterministic pushdown automaton	$A \rightarrow \gamma$
Type-3	Regular	Finite state automaton	$A \rightarrow a$ and $A \rightarrow aB$

- Most programming languages are generated by Type-2 rules.
- Trade-off between size of language & capacity to parse it.