Reflection and its use

from science to meditation

Languages

Alphabets

An *alphabet* Σ is a set of symbols

A *word* over Σ is a finite string of elements of Σ

Example

$$\Sigma_{ab} = \{a, b\}$$

Then $abba$ is a word over Σ_{ab}

$abracadabra$ is not a word over Σ_{ab}

Notation

Σ^* collection of words over Σ

$abba \in \Sigma_{ab}^*$

$abracadabra \notin \Sigma_{ab}^*$

Words

Let $\Sigma_{01} = \{0, 1\}$

Then Σ_{ab} and Σ_{01} are *isomorphic*

Enumeration of Σ_{01}^*:

$$\begin{array}{ll} 0 \text{ elements} & `' \\ 1 \text{ element} & 0, 1 \\ 2 \text{ elements} & 00, 01, 10, 11 \\ 3 \text{ elements} & 000, 001, 010, 011, 100, 101, 110, 111 \\ \ldots & \end{array}$$

The empty string is also denoted by ϵ

In biology the alphabets

$$\Sigma_{acgt} = \{A, C, G, T\} \text{ and } \Sigma_{acgu} = \{A, C, G, U\}$$

play an important role

Languages

Let Σ be an alphabet

A *language* over Σ is a collection L of words in Σ^*

Notation: $L \subseteq \Sigma^*$

The strings of a, b's with

an even number of a's
an odd number of b's

is a language L_{eo} over Σ_{ab}

For example

$$\begin{aligned} abababa &\in L_{eo} \\ ababa, abba &\notin L_{eo} \end{aligned}$$

Hofstadter's MU puzzle

Let $\Sigma_H = \{\mathsf{I}, \mathsf{M}, \mathsf{U}\}$

We generate the following language L_H over Σ_H

axiom	MI
rules	$x\mathsf{I} \Rightarrow x\mathsf{IU}$ $\mathsf{M}x \Rightarrow \mathsf{M}xx$ $x\mathsf{III}y \Rightarrow x\mathsf{U}y$ $x\mathsf{UU}y \Rightarrow xy$

This means that by definition MI in L_H

if $x\mathsf{I}$ in L_H, then also $x\mathsf{IU}$
if $\mathsf{M}x$ in L_H, then also $\mathsf{M}xx$
if $x\mathsf{III}y$ in L_H, then also $x\mathsf{U}y$
if $x\mathsf{UU}y$ in L_H, then also xy

Is the following true or not true:

$$\mathsf{MU} \text{ in } \Sigma_H?$$

More languages and their coding

Let $\Sigma = \{a, b\}$. Define the following languages over Σ.

(i) $L_1 = \{w \mid w$ starts with a and ends with $b\}$. Then

$$ab, abbab \in L_1, \text{ but } \epsilon, abba, bab, ba \notin L_1.$$

L_1 is coded by $a(a \cup b)^*b$.

(ii) $L_2 = \{w \mid abba$ is part of $w\}$. Then

$$abba, abbab \in L_2, \text{ but } \epsilon, ab, bab \notin L_2.$$

L_2 is coded by $(a \cup b)^*abba(a \cup b)^*$.

(iii) $L_3 = \{w \mid aa$ is not part of $w\}$. Then

$$\epsilon, abba, abbab \in L_3, \text{ but } aa, babaa \notin L_3.$$

L_3 is coded by $((b \cup ab)^*(a \cup \epsilon))$.

Regular languages

Let Σ be an alphabet.

(i) The *regular expressions* over Σ are defined as follows

$$\mathtt{re} := \emptyset \mid \epsilon \mid s \mid (\mathtt{re}.\mathtt{re}) \mid (\mathtt{re} \cup \mathtt{re}) \mid \mathtt{re}^*$$

here s is an element of Σ.

(ii) For a regular expression e the language $L(e)$ over Σ is defined

$$\begin{aligned} L(\emptyset) &= \emptyset \\ L(\epsilon) &= \{\epsilon\} \\ L(s) &= \{s\} \\ L(e_1 e_2) &= L(e_1)L(e_2) \\ L(e_1 \cup e_2) &= L(e_1) \cup L(e_2) \\ L(e^*) &= L(e)^* \end{aligned}$$

(iii) L over Σ is called *regular* if $L = L(e)$ for some $e \in \mathtt{re}$.

Context-free languages

Production system (grammar) over $\Sigma = \{a, b\}$.

$$\begin{array}{lll} S & \rightarrow & \epsilon \\ S & \rightarrow & aSb \end{array}$$

also written as

$$S \rightarrow \epsilon \mid aSb$$

The productions can be depicted as follows.

$$\begin{array}{l} S \rightarrow \epsilon \\ S \rightarrow aSb \rightarrow ab \\ S \rightarrow aSb \rightarrow aaSbb \rightarrow aabb \\ S \rightarrow aSb \rightarrow aaSbb \rightarrow aaaSbbb \rightarrow aaabbb \end{array}$$

$$L_5 = \{\epsilon, ab, aabb, aaabbb, a^4b^4, \ldots, a^nb^n, \ldots\},$$

also written as

$$L_5 = \{a^nb^n \mid n \geq 0\}.$$

Palindromes

Let $\Sigma = \{a, b\}$. For $w \in \Sigma^*$, let $w^\vee$ be the *reverse* word

$$\begin{array}{rcl} \epsilon^\vee & = & \epsilon; \\ (ws)^\vee & = & s(w^\vee). \end{array}$$

A *palindrome* is a word w such that $w = w^\vee$.

$$L_P = \{w \mid w \text{ is a palindrome}\}$$

For example

$$abba, bab, a, \epsilon \in L_P, \text{ but } abb, abab \notin L_P.$$

Context-free grammar for L_P

$$\boxed{S \rightarrow \epsilon \mid aSa \mid bSb}$$

Part of English

$$
\begin{aligned}
S = \langle sentence\rangle &\rightarrow \langle noun-phrase\rangle\langle verb-phrase\rangle.\\
\langle sentence\rangle &\rightarrow \langle noun-phrase\rangle\langle verb-phrase\rangle\langle object-phrase\rangle.\\
\langle noun-phrase\rangle &\rightarrow \langle name\rangle \mid \langle article\rangle\langle noun\rangle\\
\langle name\rangle &\rightarrow John \mid Jill\\
\langle noun\rangle &\rightarrow bicycle \mid mango\\
\langle article\rangle &\rightarrow a \mid the\\
\langle verb-phrase\rangle &\rightarrow \langle verb\rangle \mid \langle adverb\rangle\langle verb\rangle\\
\langle verb\rangle &\rightarrow eats \mid rides\\
\langle adverb\rangle &\rightarrow slowly \mid frequently\\
\langle adjective-list\rangle &\rightarrow \langle adjective\rangle\langle adjective-list\rangle \mid \epsilon\\
\langle adjective\rangle &\rightarrow big \mid juicy \mid yellow\\
\langle object-phrase\rangle &\rightarrow \langle adjective-list\rangle\langle name\rangle\\
\langle object-phrase\rangle &\rightarrow \langle article\rangle\langle adjective-list\rangle\langle noun\rangle
\end{aligned}
$$

Jill frequently eats a juicy yellow mango.

Other classes of languages

Context-sensitive languages are generated by rules like

$$uXv \quad \rightarrow \quad uwv$$

with $w \neq \epsilon$

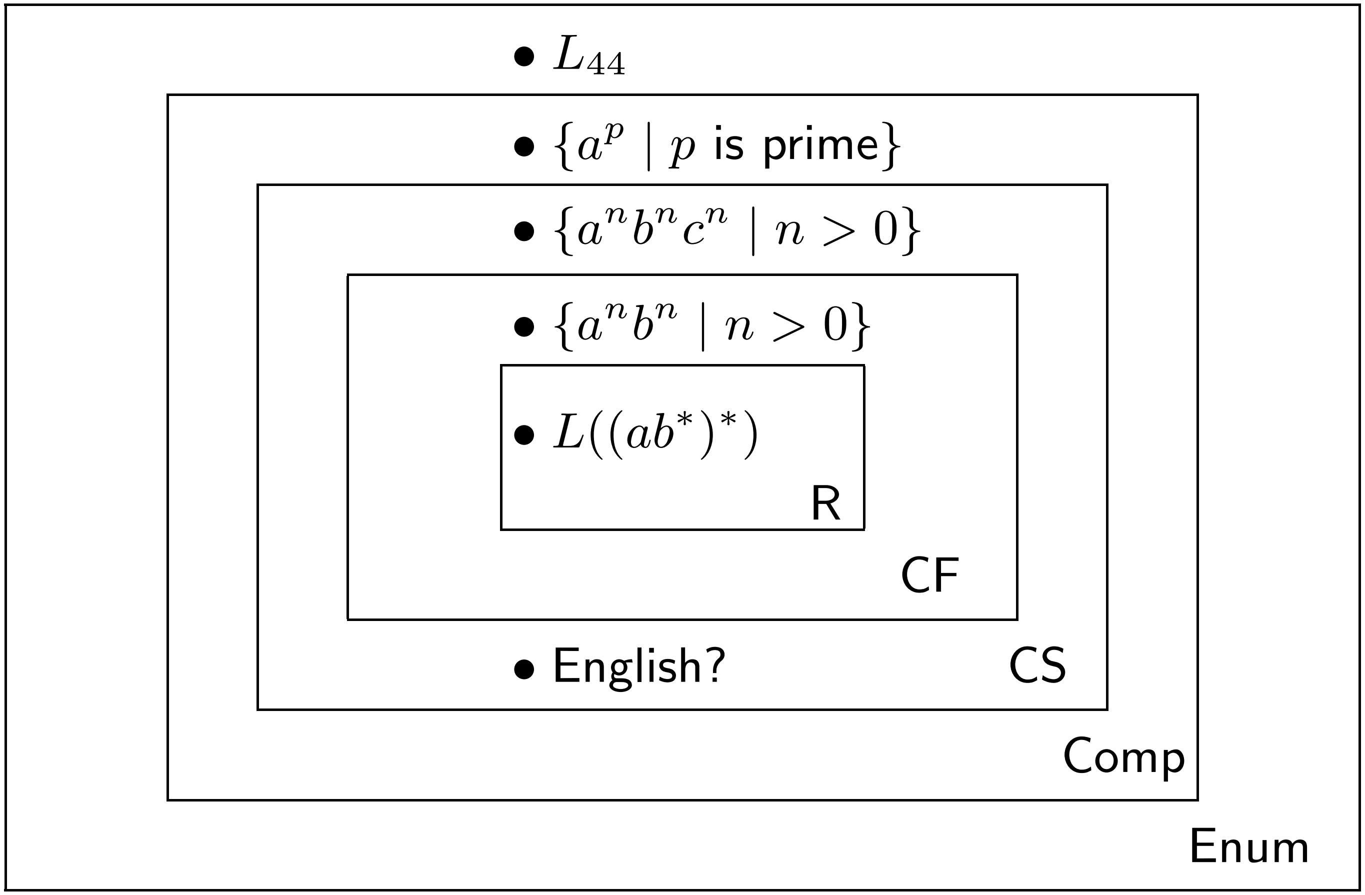
Turing enumerable languages are generated by rules like

$$uXv \quad \rightarrow \quad uwv$$

A language $L \subseteq \Sigma^*$ is *computable* if both L and $\Sigma^* - L$ are enumerable.

The Chomsky hierarchy

Let R, CF, CS, Comp, Enum denote respectively the regulae, context-free, context-sensitive, computable and enumerable languages. Then R $\subseteq$ CF $\subseteq$ CS $\subseteq$ Comp $\subseteq$ Enum.



The Chomsky hierarchy

Reflection and classes of languages

One uses the regular expressions to describe the regular languages.
These expressions do not form a regular language (but a CF one).
There is no other way to arrange this.

V. Capretta: R cannot be made into a reflexive domain.

The same also holds for Comp and probably also for CF.

The classes Enum (and CS) can be described by themselves.

There exist languages L_U, L_C in Enum such that for $c{\in}\Lambda_C$

$$L_c = \{w{\in}\Sigma^* \mid cw{\in}L_U\}$$

are exactly the languages in Enum.

Reflection and a particular language

English can describe itself.

This form of reflection and the one on previous slide are concerned with different domains.

The reflection for English is concerned with the domain sentences over the Roman alphabet.

The reflection for the class Enum is concerned with as domain a class of languages.