2. Languages

Alfabet, word

An *alphabet* Σ is a (finite) set of symbols

$$\begin{array}{llll} \text{Examples} & \Sigma_1 & = & \{a\} \\ & \Sigma_2 & = & \{0,1\} \\ & \Sigma_3 & = & \{A,C,G,T\} \\ & \Sigma_4 & = & \{a,b,c,d,\ldots,x,y,z\} \\ & \Sigma_5 & = & \{s \mid s \text{ is an ascii symbol}\} \\ & \Sigma_6 & = & \{\ldots\} \end{array}$$

A *word* (string) over Σ is a finite list of elements from Σ
The set Σ^* consists of all words over Σ

Inductive generation of words

$$\begin{array}{rcll} & & \epsilon \in \Sigma^*, & \epsilon \text{ is the empty word} \\ w \in \Sigma^* & \Rightarrow & ws \in \Sigma^*, & \text{for all } s \in \Sigma \end{array}$$

Language

A *language* over Σ is a subset of Σ^*, notation $L \subseteq \Sigma^*$

Examples

$$L_1 = \{w{\in}\{a,b\}^* \mid abba \text{ does occur as substring of } w\}$$
$$L_2 = \{w{\in}\{a,b\}^* \mid abba \text{ does not occur as substring of } w\}$$

Operations on words

$$u{\in}\Sigma^*,\ s{\in}\Sigma \quad\Rightarrow\quad us{\in}\Sigma^*$$
$$u{\in}\Sigma^*,\ v{\in}\Sigma^* \quad\Rightarrow\quad u+v{\in}\Sigma^* \qquad \textit{concatenation}$$
$$u{\in}\Sigma^*,\ n{\in}\mathbb{N} \quad\Rightarrow\quad u^n{\in}\Sigma^*$$

Inductive definitions

$$u + \epsilon = u \qquad u^0 = \epsilon$$
$$u + (vs) = (u+v)s \qquad u^{k+1} = u^k u$$

write $u + v$ as uv

Operation on languages (easy!)

$$L_1 \subseteq \Sigma^*,\ L_2 \subseteq \Sigma^* \quad \Rightarrow \quad L_1 \cup L_2 \subseteq \Sigma^*$$

$$L_1 \subseteq \Sigma^*,\ L_2 \subseteq \Sigma^* \quad \Rightarrow \quad L_1 L_2 \subseteq \Sigma^*$$

$$L \subseteq \Sigma^* \quad \Rightarrow \quad L^* \subseteq \Sigma^*$$

$$\begin{aligned}
L_1 \cup L_2 &= \{w \mid w \in L_1 \text{ or } w \in L_2\} \\
L_1 L_2 &= \{w_1 w_2 \mid w_1 \in L_1 \ \&\ w_2 \in L_2\} \\
L^0 &= \{\epsilon\} \\
L^{n+1} &= L^n L \\
L^* &= \bigcup_{n \in \mathbb{N}} L^n = L^0 \cup L^1 \cup L^2 \cup \ldots \\
&\neq \{w^* \mid w \in L\}
\end{aligned}$$

Regular expressions & languages over Σ

Let $\Sigma = \{a, b\}$. Then $a(ba)^*bb$ is a *regular expression* denoting

$$\begin{aligned} L &= \{a(ba)^n bb \mid n{\in}\mathbb{N}\} \\ &= \{abb, ababb, abababb, ababababb, \ldots, a(ba)^n bb, \ldots\} \end{aligned}$$

For general Σ the regular expressions over Σ are generated by

$$\boxed{\texttt{rexp}_\Sigma ::= \emptyset \mid \epsilon \mid \texttt{s} \mid \texttt{rexp}_\Sigma\ \texttt{rexp} \mid \texttt{rexp}_\Sigma \cup \texttt{rexp}_\Sigma \mid \texttt{rexp}_\Sigma^*}$$

with $\texttt{s}{\in}\Sigma$

This means $\emptyset{\in}\texttt{rexp}_\Sigma$, $\epsilon{\in}\texttt{rexp}_\Sigma$, and $s{\in}\texttt{rexp}_\Sigma$ for $s{\in}\Sigma$. Moreover

$$\begin{aligned} e_1, e_2{\in}\texttt{rexp}_\Sigma &\Rightarrow (e_1 \cup e_2){\in}\texttt{rexp}_\Sigma \\ e_1, e_2{\in}\texttt{rexp}_\Sigma &\Rightarrow (e_1 e_2){\in}\texttt{rexp}_\Sigma \\ e{\in}\texttt{rexp}_\Sigma &\Rightarrow e^*{\in}\texttt{rexp}_\Sigma \end{aligned}$$

Regular languages

For a regular expression e over Σ we define the language $L(e) \subseteq \Sigma^*$:

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

A language L is called *regular* if $L = L(e)$ for some $e \in \texttt{rexp}$

$L = \{w \mid bb \text{ occurs in } w\}$ over $\Sigma = \{a, b\}$ is regular:

$$L = L((a \cup b)^* bb (a \cup b)^*)$$

Also $L' = \Sigma^* - L = \{w \mid bb \text{ does not occur in } w\}$:

$$L' = L(a^*(baa^*)^* \cup a^*(baa^*)^* b)$$

Context-free languages

A *context free language* L over Σ is being generated by productionrules of the vorm

$$X \to w$$

with $X \in V$ and $w \in (\Sigma \cup V)^*$. The elementens of V are called auxiliary. There is an $S \in V$ (start). The language L is generated by the rules obtained via the relatie $\Rightarrow$ defined as follows (we have $u, v, w, x, y \in (\Sigma \cup V)^*$):

$$\begin{aligned} X \to w \quad &\Rightarrow \quad xXy \Rightarrow xwy \\ u \Rightarrow v,\ v \Rightarrow w \quad &\Rightarrow \quad u \Rightarrow w \end{aligned}$$

Finally we have

$$L = \{w \in \Sigma^* \mid S \Rightarrow w\}.$$

Notation. $X \to w_1 \mid w_2$ abbreviates

$$\begin{aligned} X &\to w_1 \\ X &\to w_2 \end{aligned}$$

$\boxed{S \to \epsilon \mid aSb}$ yields $L = \{a^n b^n \mid n \in \mathbb{N}\}$.

$\boxed{S \to \epsilon \mid a \mid b \mid aSa \mid bSb}$ yields $L = \{w \mid w \text{ is a palindrome}\}$

Small part of Engels as CF language

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

Jill frequently eats a juicy yellow mango belongs to this language

Context-sensitive and enumerable languages*

The *context sensitive languages* start with production rules of the form

$$uXv \to uwv,$$

with $w \in \Sigma^*$ not ϵ and $u, v \in (\Sigma \cup V)^*$ arbitrary.

For the *enumerable languages* production rules are of the form

$$uXv \to uwv,$$

with $u, v \in (\Sigma \cup V)^*$ and $w \in \Sigma^*$ arbitrary.

For *unrestricted languages* the production rules are of the form

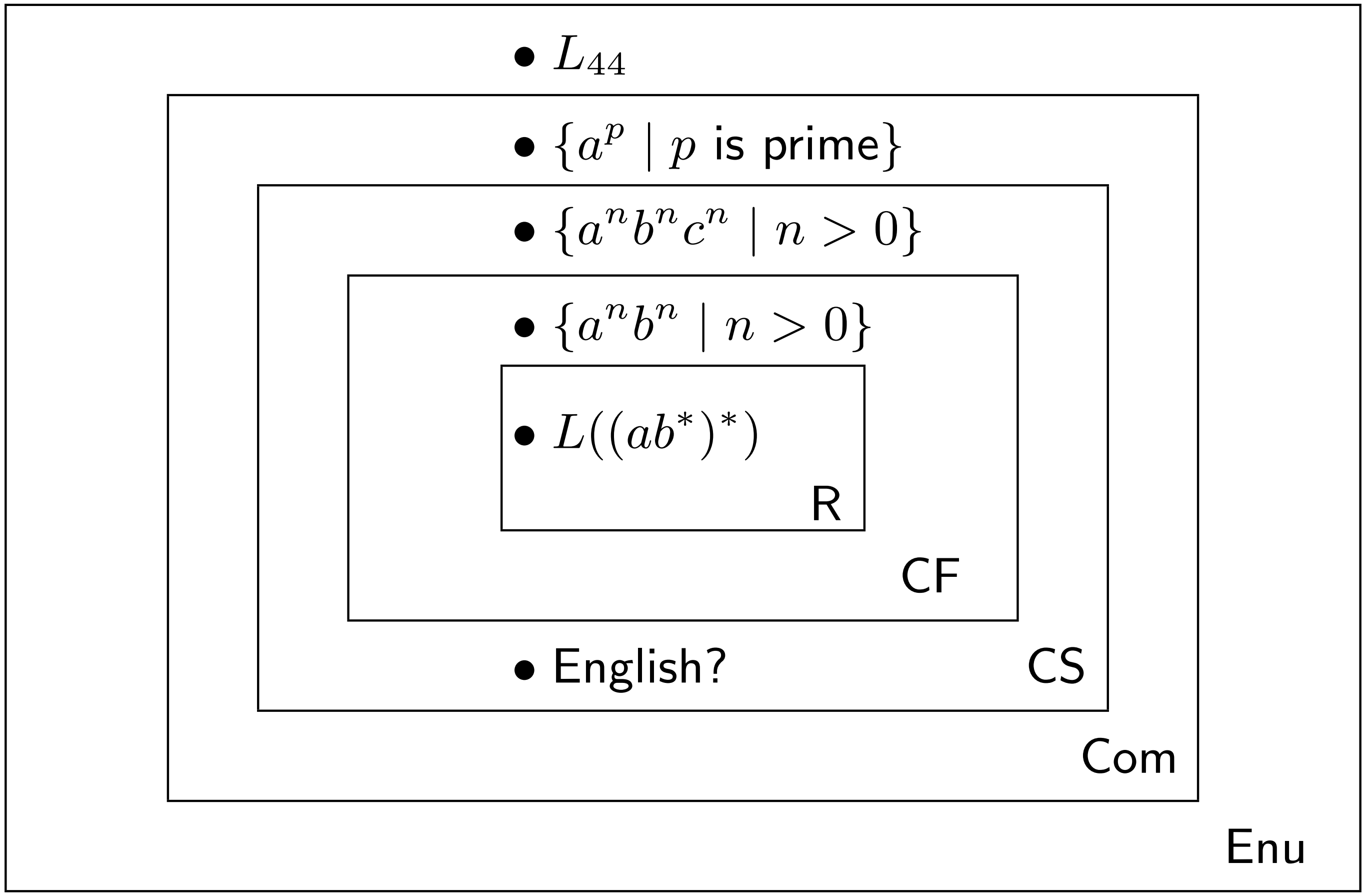
$$u \to v,$$

with $u, v \in (\Sigma \cup V)^*$.

These are equivalent to the enumerable languages.

Chomsky hierarchy*

A languages is called computable if both L and $\overline{L} = \Sigma^* - L$ are enumerable. Let R, CF, CS, Com, Enu be notations for the regular, context-free, context-sensitive, computable and enumerable languages, respectively. Then R $\subseteq$ CF $\subseteq$ CS $\subseteq$ Com $\subseteq$ Enu.



The Chomsky hierarchy