Concluding remarks for Part One
Outlook on Part Two

Regular languages

– Finite automata, regular expressions and right-linear grammars

– Pumping Lemma and Myhill-Nerode Theorem

Contex-free languages

– Non-deterministic push down automata and context-free grammars

– Pumping Lemma for context-free languages

Decidable and recursively enumerable languages

– Turing machines [and unrestricted grammars]

– Halting Problem and other undecidable problems

Language	**regular**	**context-free**	**semi-decidable/r.e.**
Grammar	right-linear	context-free	unrestricted
rules	$A \to wB$, $A \to w$	$A \to \alpha$	$\alpha \to \beta$
Machine	DFA or NFA	NPDA	Turing machine
memory	finite	finite + one stack	unrestricted
Other Descr^{n}	Regular expression Myhill-Nerode-Thm		λ-calculus μ-recursive funct^{s} Hugs, Oberon, . . .
Example	$\{ w : w \text{ contains } 10 \}$ $\{ 0^n 1^m : n, m \geq 0 \}$	$\{ w\, w^R \}$ $\{ 0^n 1^n : n \geq 0 \}$	$\{ 0^n 1^m 0^{n+m} : n \geq 0 \}$ $\{ Mx : M \text{ halts on } x \}$ HP
Counterexample	$\{ 0^n 1^n : n \geq 0 \}$	$\{ 0^n 1^n 0^n : n \geq 0 \}$ $\{ w\, w \}$	$\{ M : M \text{ halts on every } x \}$ TP
Application	Tokenizer Model checker	Parser	General computing

The Pumping Lemma for context-free languages

Pumping Lemma for CFGs. If L is a context free language, then there is a number p – the *pumping length* – such that if $w \in L$ of length at least p, then w may be divided into five pieces, $w = u\,x\,y\,z\,v$, satisfying:

(i) for each $i \geq 0$, $u\,x^i\,y\,z^i\,v \in L$

(ii) $|xz| > 0$

(iii) $|x\,y\,z| \leq p$.

This can be used to show that the following languages are not context-free

- $\{\,w\,w : w \in \{0,1\}\,\}$ **Idea**: use $w = 0^p\,1^p\,0^p\,1^p$
- $\{\,0^n 1^n 0^n : n \geq 0\,\}$ **Idea**: use $w = 0^p\,1^p\,0^p$

Regular Operations and Context-Free Languages

Theorem: Context-free languages are closed under the regular operations: union, concatenation and star.

Proof idea. Let $G_1 = (\Gamma_1, \Sigma, \mathcal{R}_1, S_1)$ and $G_2 = (\Gamma_2, \Sigma, \mathcal{R}_2, S_2)$ be context free grammars with $\Gamma_1 \cap \Gamma_2 = \emptyset$. Consider the following context free grammars

$$\begin{aligned}
G_{\text{union}} &= (\Gamma_1 \cup \Gamma_2 \cup \{\,S\,\}, \Sigma, \mathcal{R}_1 \cup \mathcal{R}_2 \cup \{\,S \to S_1, S \to S_2\,\}, S) \\
G_{\text{concat}} &= (\Gamma_1 \cup \Gamma_2 \cup \{\,S\,\}, \Sigma, \mathcal{R}_1 \cup \mathcal{R}_2 \cup \{\,S \to S_1\,S_2\,\}, S) \\
G_{\text{star}} &= (\Gamma_1 \cup \{\,S\,\}, \Sigma, \mathcal{R}_1 \cup \{\,S \to S_1\,S, S \to \epsilon\,\}, S)
\end{aligned}$$

where S is a fresh variable. Then

$$\begin{aligned}
L(G_{\text{union}}) &= L(G_1) \cup L(G_2) \\
L(G_{\text{concat}}) &= L(G_1) \cdot L(G_2) \\
L(G_{\text{star}}) &= L(G_1)^*
\end{aligned}$$

Intersection and context-free languages

1. Context-free languages are not closed under intersection.
2. The intersection of a context-free language with a regular set is context-free.

Proof Idea

1. Both $A = \{\, 0^n\, 1^n\, 0^m : n, m \geq 0 \,\}$ and $B = \{\, 0^n\, 1^m\, 0^m : n, m \geq 0 \,\}$ are context-free. But $A \cap B = \{\, 0^n\, 1^n\, 0^n : n \geq 0 \,\}$ is not context-free.
2. Let A be a regular set accepted by NFA $M = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and let B be a context-free language accepted by NPDA $N = (Q_2, \Sigma, \Gamma, \delta_2, q_2, \perp, F_2)$.
   Define the "product"-automaton $P = (Q_1 \times Q_2, \Sigma, \Gamma, \delta, (q_1, q_2), \perp, F_1 \times F_2$ where δ is given by
   $$\delta((r,t), a, A) = \{\, ((r', t'), \alpha) : r' \in \delta_1(r, a), (t', \alpha) \in \delta_2(t, a, A) \,\}$$
   then P is a NPDA that accepts $A \cap B$.

Complementation and context-free languages

Context-free languages are not closed under complementation.

Proof Idea The set $A = \{\, w\, w : w \in \{0, 1\} \,\}$ is not context-free, but its complement $B = \{0, 1\}^* \setminus A$ is context-free, as it is generated by the grammar $(\{\, S, A, B, C \,\}, \{\, 0, 1 \,\}, \mathcal{R}, S)$

whith rule set $\mathcal{R} = \{\; S \;\rightarrow\; AB \mid BA \mid A \mid B,$
$$A \rightarrow CAC \mid 0,$$
$$B \rightarrow CBC \mid 1,$$
$$C \rightarrow 0 \mid 1 \;\}$$

This grammar generates

all strings of odd length starting with productions $S \rightarrow A$ or $S \rightarrow B$ or

strings of the form $x0yu1v$ or $u1vx0y$ where $x, y, u, v \in \{0, 1\}^*$, $|x| = |y|$ and $|u| = |v|$. None of these strings can be of the form $w\, w$.