1. Reading: D. Kozen *Automata and Computability*, Lectures 15, 16
J. Hopcroft and J. Ullman *Introduction to Automata Theory, etc.*, section 3.4.

2. The main message of this lecture:

The first really deep theorem of the course: for every regular language $A$ there exists a unique minimum state DFA accepting $A$. Moreover, such an automaton can be obtained from any DFA accepting $A$ by pruning out inaccessible states and applying the minimization algorithm (Myhill-Nerode Theorem).

Imagine that two teams have different ideas of how to write a DFA accepting the same language $A$ and eventually come with two different solutions $M_1$ and $M_2$. Naturally, we are interested in a DFA $M$ having the fewest number of states possible and we decide to apply the minimization algorithm. Shall we apply minimization to both $M_1$ and $M_2$? May be our competitor will do even better and come with truly ingenious $M_3$? The wonderful Myhill-Nerode Theorem clarifies the picture immensely: in all of those cases we end up with the same minimum state DFA $M$!

**Definition.** Two DFA are *isomorphic* if one of them can be obtained from another by renaming of states. Here is the ‘official’ formulation: an *isomorphism* $f$ of DFA $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ and $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ is a one-to-one and onto mapping from $Q_M$ to $Q_N$ preserving ‘start’, ‘accept’ and the transition function: $f(s_M) = s_N, p \in F_M \iff f(p) \in F_N, f(\delta_M(p, a)) = \delta_N(f(p), a)$. Isomorphic automata have equal number of states, similar ‘start’ and ‘accept’ states, identical transition functions, and accept the same regular languages.

**Definition.** An *index* of an equivalence relation $\approx$ on $Q$ is the number of equivalence classes with respect to $\approx$. An equivalence relation $\approx_1$ is a *finer* than an equivalence relation $\approx_2$ ($\approx_2$ is coarser than $\approx_1$) if every equivalence class of $\approx_1$ is entirely contained in some equivalence class of $\approx_2$: $x \approx_1 y \Rightarrow x \approx_2 y$. An equivalence relation $\approx$ refines a set $R$ if every equivalence class of $\approx$ is either entirely in $R$ or entirely in $\sim R$: $x \approx_R y \Rightarrow (x \in R \iff y \in R)$. An equivalence relation $\approx$ on $\Sigma^*$ is a right congruence if $x \approx y \Rightarrow xz \approx yz$ for each strings $x, y, z \in \Sigma^*$.

**Definition.** Let $R \subseteq \Sigma^*$. We define an equivalence relation $\equiv_R$ on $\Sigma^*$ as

$x \equiv_R y \iff \forall z \in \Sigma^* (xz \in R \iff yz \in R)$.

**Example 14.1.** $R = \{a^{2n} \mid n \geq 0\} = \{\epsilon, aa, aaaa, \ldots\}$. Here $\equiv_R$ has index 2, i.e. there are only two equivalence classes: $[\epsilon] = \{\epsilon, aa, aaaa, \ldots\} = R$ and $[a] = \{a, aaa, aaaaa, \ldots\} = Ra$.

**Example 14.2.** $R = \{a^{n^2} \mid n \geq 0\} = \{\epsilon, a, a^4, a^9, \ldots\}$. Here $\equiv_R$ is of infinite index, i.e. there are infinitely many equivalence classes here. Indeed, it is easy to check that any two elements of $R$ are not equivalent and hence generate distinct equivalence classes. For example, $[a] \neq_R [aaaa]$, since $a \cdot a = a^4 \in R$, but $aaaa \cdot aaa = a^{10} \notin R$.

Note that $R$ from 14.1 is regular whereas $R$ from 14.2 is not.
Lemma 14.3. $\equiv_R$ is a right congruence refining $R$ and is the coarsest such relation on $\Sigma^*$. 

Proof. Right congruence: Let $x \equiv_R y$, i.e. $\forall z \in \Sigma^* (xz \in R \iff yz \in R)$. Then $xz \equiv_R yw$ for any string $w$. Indeed, for any string $z$

$$(xz)w \equiv_R xz \iff yz \iff (yw)z \in R.$$ 

Refines $R$: take $z = e$ in the definition of $x \equiv_R y$ and get $(x \in R \iff y \in R)$. $\equiv_R$ is the coarsest: let $\equiv$ is a right congruence refining $R$. Then

$$x \equiv y \Rightarrow \forall z (xz \equiv yz) \Rightarrow \forall z (xz \in R \iff yz \in R) \Rightarrow x \equiv_R y.$$ 

Theorem 14.4 (Myhill-Nerode Theorem) Let $R \subseteq \Sigma^*$. Then $R$ is regular if and only if the relation $\equiv_R$ is of finite index.

Proof. Let $R = L(M)$ for some DFA $M$. Define an equivalence relation $x \equiv_M y$ on strings over $\Sigma$ as $\delta(s, x) = \delta(s, y)$. $\equiv_M$ is a right congruence: $x \equiv_M y \Rightarrow \delta(s, x) = \delta(s, y) \Rightarrow \delta(s, x^i) = \delta(s, y^i) \Rightarrow xz \equiv_M yz$. It is also clear that $\equiv_M$ refines $R$: $x \equiv_M y \Rightarrow \delta(s, x) = \delta(s, y) \Rightarrow (\delta(s, x) \in F \iff \delta(s, y) \in F) \Rightarrow (x \in R \iff y \in R)$. By lemma 14.3, $\equiv_R$ is coarser than $\equiv_M$. In particular, $\equiv_R$ has less equivalence classes than $\equiv_M$. Note that $\equiv_M$ is of finite index, since the number of equivalence classes for $x \equiv_M y$ does not exceed the number of states in $M$. Therefore $\equiv_R$ is also of finite index not exceeding the number of states in $M$.

Let now $\equiv_R$ be of finite index. Define $M^R = (Q, \Sigma, \delta, s, F)$ such that $Q = (R/\equiv_R)$ (a finite set of equivalence classes with respect to $\equiv_R$), $\delta([x], a) = [xa]$, $s = [e]$, $F = \{[x] | x \in R\}$. We claim that $\delta([x], y) = [xy]$. Induction on $|y|$. The induction base is secured by the definition of $\delta$ above. The induction step: $\delta([x], ya) = \delta([x], y, a) = \delta([xy], a)$ (by the induction hypothesis) $= [xya]$. Claim: $R = L(M^R)$. Indeed

$$x \in L(M^R) \iff \delta([x], x) \in F \iff [x] \in F \iff [x] \in F \iff x \in R.$$ 

Corollary 14.5 $M^R$ has the fewest number of states among all DFAs accepting $R$.

Corollary 14.6 The collapsing minimization algorithm returns a DFA isomorphic to $M^R$.

Proof. Let $N/\approx = (Q', \Sigma, \delta', s', F')$ be the collapsed automaton accepting $R$, and $M^R$ as in Theorem 14.4. We define an isomorphism $f$ from $M_R$ to $N/\approx$: $f([x]) = \hat{\delta}(s', x)$. The mapping $f$ is one-to-one. Indeed, suppose $f([x]) = f([y])$, i.e. $\hat{\delta}(s', x) = \hat{\delta}(s', y)$. Then $\hat{\delta}(\hat{\delta}(s', x), z) = \hat{\delta}(\hat{\delta}(s', y), z), \hat{\delta}(s', xz) = \hat{\delta}(s', yz), xz \in R \iff yz \in R$, therefore $[x] = [y]$. $f$ is onto, since each state $q' \in Q'$ in $N/\approx$ is accessible: there exists $x$ such that $q' = \hat{\delta}(s', x)$. Start state: $f(s) = f([e]) = \hat{\delta}(s', e) = s'$. Accept states: $[x] \in F \iff x \in R$ (above) $\iff \hat{\delta}(s', x) \in F'$ (since $N$ accepts $R$) $\iff f([x]) \in F'$ (definition of $f$). Let us do the transition function. $f(\delta([x], a)) = f([xa]) = \hat{\delta}(s', xa) = \hat{\delta}'(\hat{\delta}(s', x), a) = \hat{\delta}'(f([x]), a)$.

Example 14.7 The Myhill-Nerode automaton for $R = \{a^{2n} \mid n \geq 0\}$ from Example 14.1 has two states $[e] = R$ and $[a] = Ra$, $s = [e], F = \{R\} = \{[e]\}, \delta([e], a) = [a], \delta([a], a) = [e]$. 

Problem 14.1 #53 from Kozen p. 329.

Problem 14.2 #55a from Kozen p. 329.