In this exercise set we refer to two textbooks:

[Kozen] Dexter C. Kozen. *Automata and Computability.* Springer Verlag 1997.

[Hopcroft&Ullman] John E. Hopcroft and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley 1979.

1 Basic Concepts

1.1 Let w be the string *abcde*.

a) Give all prefixes of w.

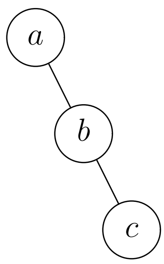
b) Give all suffixes of w.

1.2 Suppose $L_1 = \{carl, hugh, paul\}$ and $L_2 = \{smith, jones\}$. Enumerate the strings which belong to the language $L_3 = L_1L_2$ (i.e. $L_3 = \{xy \mid x \in L_1, y \in L_2\}$).

1.3 If L is a language, then L^n denotes the language which is obtained by concatenating L n times (i.e. $L^0 = \{\epsilon\}$ and for $n > 0$, $L^n = LL^{n-1}$). Furthermore, L^* denotes $\cup_{n=0}^{\infty} L^n$.

Let $L_1 = \{mor, far\}$ and $L_2 = \{s\}$. Give examples of strings in the language $(L_1^2 L_2)^* L_1 \cup L_1^2$.

1.4 The *depth* of a node v in a tree T is defined as follows. If v is the root node of T, then the depth of v is 0. Otherwise v belongs to a subtree T' of the root of T (i.e. T' is a tree such that the root of T is the parent of the root of T'), and the depth of v in T is defined to be one more than the depth of v in T'. As an example, the depth of the node c in the tree below is 2.



The *height* of a tree is the largest depth of a node in the tree. A tree is called a binary tree if every its node has either no children or exactly two children. (So the tree in the diagram is not binary). Suppose T is a binary tree of height k. Show that T has n nodes, where n satisfies the condition $2k + 1 \leq n \leq 2^{k+1} - 1$.

1.5 If Σ is an alphabet, then Σ^* denotes the language which comprises all strings which can be formed by using the symbols in Σ. For $x \in \Sigma^*$, let x^R denote x reversed and be defined recursively:

1. If $x = \epsilon$, then $x^R = \epsilon$
2. If $x = ay$ for some $a \in \Sigma$ and $y \in \Sigma^*$, then $x^R = y^R a$

Let $|x|$ denote the length of a string x. Give a recursive definition of the length of a string and then show $|x| = |x^R|$ for all strings $x \in \Sigma^*$.

1.6 The set of all subsets of a set A is called the power set of A. It is denoted by 2^A.

a) Give 2^A for $A = \{a, b, c\}$.

b) Show by induction that the number of elements in 2^A is 2^n if the number of elements in A is n.

1.7 Let Σ be an alphabet and $L \subseteq \Sigma^*$ a language. Consider the relation $R_L \subseteq \Sigma^* \times \Sigma^*$ defined by: xR_Ly if and only if for all $z \in \Sigma^*$, $xz \in L \iff yz \in L$.

a) Show that R_L is an equivalence relation.

b) Give the equivalence classes of R_L for $L = \{(01)^n \mid n \geq 0\}$ and $\Sigma = \{0, 1\}$.

c) Give the equivalence classes of R_L for $L = \{0^n1^n \mid n \geq 1\}$ and $\Sigma = \{0, 1\}$.

d) Give the equivalence classes of R_L for $L = \{0^n10^m \mid n \geq 0, m \geq 0\}$ and $\Sigma = \{0, 1\}$.

If x is a string, then x^n denotes the concatenation of n x's. Parentheses are used for showing where a string begins and ends. They are omitted if the string consists of a single symbol. For example, $(01)^2$ is the string 0101 and $(01)^0$ is the empty string. Relation R_L is denoted in [Kozen] by $\equiv_L$.

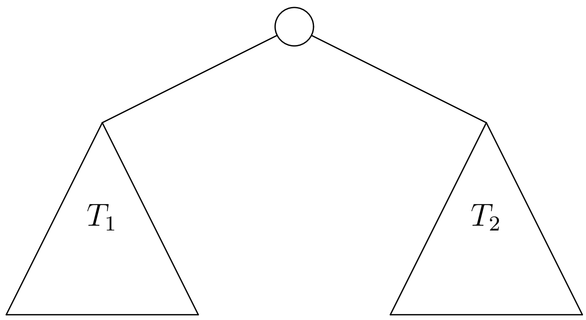
Suggested Solutions

1.4 Let $IH(k)$ be: 'the number of nodes n in a binary tree of height k satisfies the condition $2k+1 \leq n \leq 2^{k+1}-1$'. What we have to show is thus that $IH(k)$ holds for all $k \geq 0$. This is shown by induction on k.

Basis $IH(0)$ holds.
$k = 0$ implies that the number of nodes is 1, i.e. $n = 1$. $2 \cdot 0 + 1 = 1 = 2^{0+1} - 1$.

Inductive hypothesis Suppose $IH(k)$ holds for some $k \geq 0$.

Inductive step Show that $IH(k+1)$ then holds.
Consider a binary tree of height $k+1$:



One of the subtrees T_1 and T_2 must then be of height k, otherwise the height of T would not be $k+1$. (The height of the other subtree is $\leq k$). Let n_1, n_2 be the numbers of nodes of T_1 and T_2, respectively.

1. By the inductive assumption, $n_1 \leq 2^{k+1}-1$ and $n_2 \leq 2^{k+1}-1$. Thus the total number of nodes in the tree is $n = 1 + n_1 + n_2 \leq 1 + 2(2^{k+1}-1) = 2^{(k+1)+1} - 1$.
2. By the inductive assumption, the subtree of height k has $n_i \geq 2k+1$ nodes. The other has $n_j \geq 1$ nodes. Thus the total number of nodes in the tree is $n = 1 + n_i + n_j \geq 1 + (2k+1) + 1 = 2(k+1) + 1$.

1 and 2 imply that the number of nodes n in the tree satisfies $2(k+1)+1 \leq n \leq 2^{(k+1)+1} - 1$, i.e. $IH(k+1)$ holds.

By mathematical induction $IH(k)$ holds for all $k \geq 0$.

1.5 The length of a string x, $|x|$, can be defined recursively as follows.

1. If $x = \epsilon$, then $|x| = 0$.
2. If $x = ay$ for some $a \in \Sigma$ and $y \in \Sigma^*$, then $|x| = 1 + |y|$

Let $IH(k)$ be: '$|x| = k$ if and only if $|x^R| = k$, i.e. $|x| = |x^R|$'. Show that $IH(k)$ holds for all $k \geq 0$.

Basis $IH(0)$ holds.
$|x| = 0 \Leftrightarrow x = \epsilon \Leftrightarrow x^R = \epsilon \Leftrightarrow |x^R| = 0$

Inductive hypothesis Suppose $IH(k)$ holds for some $k \geq 0$.

Inductive step Show that $IH(k+1)$ then holds.
$|x| = k+1 \Leftrightarrow x = ay$ and $|y| = k$ for some $a \in \Sigma$ and $y \in \Sigma^*$. The induction hypothesis implies $|y^R| = k$ and thus we have $|x^R| = |y^R a| = k+1$.

What is missing? From the definition of $|\cdot|$ above, it does not follow immediately that the equality $|y^R a| = k+1$ really holds. We show this again by induction.

Let $IH'(k)$ be: 'if $|x| = k$ then $|xa| = k+1$, for any $x \in \Sigma^*$ and $a \in \Sigma$'.

Basis $IH'(0)$ holds.
If $|x| = 0$, then $x = \epsilon$. Thus $|xa| = |a| = 1$.

Inductive hypothesis Suppose $IH'(k)$ holds for some $k \geq 0$.

Inductive step Show that $IH'(k+1)$ then holds.
If $|y| = k+1$ then $y = bx$, where $|x| = k$. So $ya = bxa$. By the assumption, $|xa| = k+1$. Hence $|by| = |bxa| = k+2$ by the definition of $|\cdot|$.

1.7 a) In order to show that R_L is an equivalence relation, it must be shown that R_L is reflexive, symmetric and transitive.

reflexive For all $x \in \Sigma^*$, show xR_Lx.
Choose an arbitrary string $x \in \Sigma^*$. Then it is obviously true that for all $z \in \Sigma^*$ $xz \in L \Leftrightarrow xz \in L$. Thus xR_Lx.

symmetric For all $x, y \in \Sigma^*$, show $xR_Ly \Rightarrow yR_Lx$.
Choose $x, y \in \Sigma^*$ such that xR_Ly. xR_Ly iff for all $z \in \Sigma^*$, $xz \in L \Leftrightarrow yz \in L$. We conclude that for all $z \in \Sigma^*$, $yz \in L \Leftrightarrow xz \in L$, i.e. yR_Lx.

transitive For all $x, y, w \in \Sigma^*$, show $xR_Ly \wedge yR_Lw \Rightarrow xR_Lw$.
Choose $x, y, w \in \Sigma^*$ such that xR_Ly and yR_Lw, and choose $z \in \Sigma^*$ arbitrary. Suppose $xz \in L$. xR_Ly implies $yz \in L$, which in turn implies $wz \in L$. Suppose instead $xz \notin L$. xR_Ly implies $yz \notin L$, which in turn implies $wz \notin L$. Thus we may conclude $xz \in L \Leftrightarrow wz \in L$, i.e. xR_Lw.

b) The equivalence classes constitute a partitioning of the set on which the equivalence relation is defined, here Σ^*. We start by finding the equivalence class for some suitable string, e.g. ϵ. If we find that $[\epsilon]$, the equivalence class for ϵ, does not cover Σ^* (i.e., $[\epsilon]$ is a proper subset of Σ^*), we continue by choosing a new string which does not belong to $[\epsilon]$. This string is a representative of a new equivalence class. We determine this equivalence class and check whether the union of all the equivalence classes obtained thus far is equal to Σ^*. If not, a new representative is chosen, its equivalence class determined and so forth until all classes have been found.

How do we determine the equivalence class for a string x? A string y is related to x, xR_Ly, if and only if for all $z \in \Sigma^*$, $xz \in L \Leftrightarrow yz \in L$. The condition $xz \in L \Leftrightarrow yz \in L$ can be restated as $xz \in L \Rightarrow yz \in L$ and $xz \notin L \Rightarrow yz \notin L$. Given a string x, we first determine what z should look like in order that $xz \in L$. Since $yz \in L$ should hold, the forms z may take constrain the strings y which may be related to x. We then proceed to check the second half of the condition. i.e. $xz \notin L \Rightarrow yz \notin L$. For those z which satisfy $xz \notin L$, it should also be the case that $yz \notin L$. This may mean that some of the strings we found earlier do not qualify. The remaining strings thus constitute $[x]$, the equivalence class for x.

1. $[\epsilon]$
   $x = \epsilon$ and $xz = z \in L$ implies $z = (01)^n$, $n \geq 0$. If $z = (01)^n$ and $yz \in L$, it must be the case that $y = (01)^m$, $m \geq 0$ since $yz = (01)^m(01)^n = (01)^{m+n}$.
   $x = \epsilon$ and $xz = z \notin L$ implies $z \neq (01)^n$, $n \geq 0$. If $z \neq (01)^n$ and $y = (01)^m$, then $yz \notin L$ holds. Thus $[\epsilon] = \{(01)^m \mid m \geq 0\}$.
2. $[0]$
   $0 \notin [\epsilon]$. $x = 0$ and $xz = 0z \in L$ implies $z = 1(01)^n$, $n \geq 0$. If $z = 1(01)^n$ and $yz \in L$, it must be the case that $y = (01)^m 0$ since $yz = (01)^m 01(01)^n = (01)^{m+n+1}$.
   $x = 0$ and $xz = 0z \notin L$ implies $z \neq 1(01)^n$, $n \geq 0$. If $z \neq 1(01)^n$ and $y = (01)^m 0$, then $yz \notin L$ holds. Thus $[0] = \{(01)^m 0 \mid m \geq 0\}$
3. $[1]$
   $1 \notin [\epsilon] \cup [0]$. $x = 1$ implies that $1z \notin L$ for an arbitrary choice of $z \in \Sigma^*$. This means that all strings y in $[1]$ must be such that $yz \notin L$ holds for an arbitrary chosen string z. For each $y \in \Sigma^* - ([\epsilon] \cup [0])$ (i.e. those strings which does not belong to $[\epsilon]$ or $[0]$) it is the case that $yz \notin L$ regardless of how z is chosen. Thus $[1] = \Sigma^* - ([\epsilon] \cup [0])$.

c) Here we get an infinite number of equivalence classes since each string of the form 0^n, $n \geq 0$ is only related to itself. This yields the following classes:

1. $[0^n] = \{0^n\}$, $n \geq 0$
2. $[01] = \{0^n 1^n \mid n \geq 1\}$
3. $[0^{k+1}1] = \{0^{k+n}1^n \mid n \geq 1\}$, $k \geq 1$
4. $[1] = \Sigma^* - ([01] \cup (\cup_{n=0}^{\infty}[0^n]) \cup (\cup_{n=2}^{\infty}[0^n 1])$

d)
1. $[\epsilon] = \{0^n \mid n \geq 0\}$
2. $[1] = \{0^n 1 0^m \mid n \geq 0, m \geq 0\}$
3. $[11] = \Sigma^* - ([\epsilon] \cup [1])$