Fundamentele Informatica 3
Antwoorden op geselecteerde opgaven uit Hoofdstuk 8

John Martin: Introduction to Languages and the Theory of Computation (fourth edition)

Jetty Kleijn, Rudy van Vliet

Voorjaar 2012

8.6 Describe algorithms to enumerate the given sets.
a The set of all pairs (n, m) where n and m are relatively prime, positive integers.
This is a recursive set: to determine whether a pair (n, m) belongs to it one could, e.g., determine their greatest common divisor; if that is 1, then (n, m) is in, otherwise not. An enumerating algorithm could now systematically generate the set of all pairs (n, m), for instance canonically, guided by the sum of the elements: $(1, 1), (1, 2), (2, 1), (1, 3), (2, 2), (3, 1), (1, 4), (2, 3), \ldots$ and successively, for each pair just generated, determine (effectively!) whether it satisfies the condition. If and only if it does, it appears as the next element of the requested enumeration.

b The set of all strings over $\{0, 1\}$ which contain a nonnull substring of the form www.
This is again a recursive set and a similar method as above applies.

c The set $\{n \mid n \geq 0 \text{ and } \exists x, y, z \text{ integers such that } x^n + y^n = z^n\}$.
More or less as before: select an enumeration of 4-tuples (n, x, y, z), compute for each such tuple $x^n + y^n$ and z^n and compare. In case of equality, n will be the next element of the list provided it was not listed already.

8.10 Let L be a language.
We have to prove that L is accepted by a Turing machine if and only if there is Turing machine which computes a function with domain L.
The "if" direction is simple: any Turing machine T which computes a function with domain L stops successfully for all words in L and for no other words. Thus $L(T) = L$.
Conversely, assume that L is recursively enumerable and let T be a Turing machine such that $L(T) = L$. We modify T as follows. First we take care that the rightmost cell left non-blank during an accepting computation can always be found (see Example 7.21 of the book). Then we further adapt the machine by letting it erase the tape and go back to cell 0. Thus the resulting TM T' accepts exactly the same words as T, that is the words from L, and moreover any successful computation is terminated with the head on cell 0 of an empty tape. Hence T' computes the partial function $f(x) = \lambda$ if $x \in L$, and $f(x)$ undefined otherwise.

8.12 We only give the solution of the "if" direction.

Assume that there is an increasing, computable, total function $f : \Sigma^* \to \Sigma^*$ whose *range* is L. The range of f is $\{y \in \Sigma^* \mid \exists x \in \Sigma^* \text{ such that } f(x) = y\}$. Note that it follows from the definition of 'increasing' that f is injective.

Let $x_0, x_1, x_2, \ldots$ be the strings from Σ^* in canonical order, and let y be an arbitary string in Σ^*, hence, $y = x_n$ for some $n \geq 0$. We can determine whether y is in the range of f by computing $f(x_0), f(x_1), f(x_2), \ldots$, (which is possible, because f is a computable, total function) until we find a string x for which $f(x) = y$ or for which y precedes $f(x)$. Because f is increasing, we will find such x in at most $n + 1$ attempts. If $f(x) = y$, then y is in the range of f. Otherwise, y is not in the range of f.

8.17
In this exercise, unrestricted grammars are given through their productions, and the exercise is to describe the languages they generate.

a Similar to the grammars in Example 8.12 and Example 8.17, but now the terminal symbols a, b and c may end up in any order. The language generated is $\{w \in \{a, b, c\}^* \mid n_a(w) = n_b(w) = n_c(w)\}$.

b Similar to the grammar in Example 8.11, but now we have D for doubling a's and T for trebling. Thus, the language generated is $\{a^n \mid n = 2^j 3^k \text{ for some } j, k \geq 0\}$.

8.18 a Every derivation begins with the step $S \Rightarrow TD_1D_2$ after which either each of the three symbols is rewritten into Λ (hence Λ is in the language) or the production $T \rightarrow ABCT$ is applied. This production introduces A's, B's, and C's in equal numbers. The two productions $AB \rightarrow BA$ and $BA \rightarrow AB$ allow one to rearrange the A's and B's in any order, while the productions $CA \rightarrow AC$ and $CB \rightarrow BC$ let the C's be shifted to the right. The symbols D_1 and D_2 — originally at the right end of the string — are crucial for successful termination (introducing terminal symbols):
D_1 can be moved to the left, past C's with $CD_1 \rightarrow D_1C$ on the way changing B's into b's with $BD_1 \rightarrow D_1b$. Note that it is blocked by A's and a's. Since this is the only way to rewrite B's into terminal strings, it follows that, if there are any B's to the left of an A, they cannot terminate and the derivation is not successful.
D_2 can also be moved to the left, on the way changing C's into a's with $CD_2 \rightarrow D_2a$. This is the only way to let the C's successfully terminate and therefore they should have been moved to the right of the A's and B's.
The language generated by this grammar is $\{a^nb^na^n \mid n \geq 0\}$

b Replace $BD_1 \rightarrow D_1b$ by $B \rightarrow b$. Then the B can always terminate, regardless of the positions of the A's and a's.

8.19 a To start with we will have productions $S \rightarrow ABCDS \,|\, ABCD$ to generate strings with 4 types of positions in equal numbers;
$BA \rightarrow AB$, $CA \rightarrow AC$, $DA \rightarrow AD$, to move A's to the left;
$CB \rightarrow BC$, $DB \rightarrow BD$ to move B's past C's and D's to the left;
$DC \rightarrow CD$ to move C's to the left of D's.
Next we introduce auxiliary variables to force a form $A^nB^nC^nD^n$: A, B, C, D can be changed into A_1, B_1, C_1, D_1, respectively, only if they are in this order and only A_1, B_1, C_1, D_1 can terminate as a or b. Note that without this intermediate step, errors may occur because a's (and b's) occur in different parts of the string.
We define a new axiom S_1 with productions $S_1 \rightarrow A_1BCDS \,|\, A_1BCD \,|\, \Lambda$, to have a starter-$A_1$ and to generate Λ.
The subscript 1 propagates from left to right through the string but on its way it can only pass from A's to B's: $A_1A \rightarrow A_1A_1$, $A_1B \rightarrow A_1B_1$,
from B's to C's: $B_1B \rightarrow B_1B_1$, $B_1C \rightarrow B_1C_1$,
and from C's to D's: $C_1C \rightarrow C_1C_1$, $C_1D \rightarrow C_1D_1$, and $D_1D \rightarrow D_1D_1$.
Finally, we include the terminating productions $A_1 \rightarrow a$, $B_1 \rightarrow b$, $C_1 \rightarrow a$ and $D_1 \rightarrow b$.

Note that the grammar is context-sensitive (monotone), except for the production $S_1 \to \Lambda$ which can anyway be omitted if we do not care about having Λ in the language.

b Similar to **a**, but for 3 positions; note that B_1 can be replaced by a or b.
$S_1 \to A_1BCS \,|\, A_1BC \,|\, \Lambda$,
$S \to ABCS \,|\, ABC$,
$BA \to AB$, $CA \to AC$, $CB \to BC$,
$A_1A \to A_1A_1$, $A_1B \to A_1B_1$, $B_1B \to B_1B_1$, $B_1C \to B_1C_1$, $C_1C \to C_1C_1$,
$A_1 \to a$, $B_1 \to a \,|\, b$, $C_1 \to a$.

c We have to generate words consisting of a concatenation of 3 copies of the same word. To this aim we use two variables A and B to travel through the word and deposit the corresponding terminal in each of the three substrings.
$S \to LMR$,
where L, M, and R mark the beginning of each subword to be; each of them may disappear (when this happens at the wrong moment, the derivation will not be successful): $L \to \Lambda$, $M \to \Lambda$, $R \to \Lambda$;
L generates a terminal and sends a messenger: $L \to LaA \,|\, LbB$;
the messenger travels to the right: $Aa \to aA, Ab \to bA, Ba \to aB, Bb \to bB$;
until it meets M at the beginning of the second copy where it leaves its message: $AM \to MaA, BM \to MbB$;
the messenger continues to R, at the beginning of the third copy where it leaves its message and then disappears: $AR \to Ra, BR \to Rb$.

d As in **c** above, but now the messenger has to leave its message not at the beginning but at the end of the middle subword. We thus change the last four productions into: $AM \to MA, BM \to MB, AR \to aRa, BN \to bRb$.

8.23 The grammar in Example 8.11 is not context-sensitive (monotone) because of the productions $L \to \Lambda$, $DR \to R$, and $R \to \Lambda$. In order to make it context-sensitive, we include the markers L and R as subscripts in other symbols. We use a's already present in the string to "carry" the markers. This leads to the grammar:
$S \to a$ generates the shortest word a directly and
$S \to a_La_R$ produces two a's to carry L and R.
$a_L \to a_La_D$ doubles the first a and introduces D.
$a_Da \to aaa_D$ lets D pass a while doubling it.
$a_Da_R \to aaa_R$, if D meets R then the index disappears and the rightmost a is doubled.
$a_L \to a$ and $a_R \to a$ termination.

8.24 We can use the idea from Exercise 8.19(c) to find the following unrestriced grammar for $XX = \{xx \mid x \in \{a,b\}^*\}$:
$S \to LR$;
$L \to \Lambda, R \to \Lambda$;
$L \to LaA \mid LbB$;
$Aa \to aA, Ab \to bA, Ba \to aB, Bb \to bB$;
$AR \to Ra; BR \to Rb$.

It has two productions ($L \to \Lambda$ and $R \to \Lambda$) violating the condition for context-sensitiveness. In this exercise we do not need Λ as an element of the language to be generated and we can thus proceed as follows.
L is replaced by A_1 or B_1 to mark the first symbol of the left half; similarly, A_2 and B_2 rather than R are used to mark the first symbol in the right half of the word. We thus obtain the following context-sensitive grammar:
$S \to A_1A_2 \mid B_1B_2$;
$A_1 \to a,\ B_1 \to b,\ A_2 \to a,\ B_2 \to b$;
alternatively, the first marker symbol can produce an additional a or b, and send the corresponding message:
$A_1 \to A_1aA \mid A_1bB,\ B_1 \to B_1aA \mid B_1bB$;
$Aa \to aA,\ Ab \to bA,\ Ba \to aB,\ Bb \to bB$;
when the messenger arrives at the second marker, this marker produces the terminal:
$AA_2 \to A_2a,\ BA_2 \to A_2b,\ AB_2 \to B_2a,\ BB_2 \to B_2b$.

8.26
In this exercise, CSG's are to be given equivalent to the unrestricted grammars in Exercise 8.17, parts (b) and (c).

b The grammar given is not context-sensitive (monotone) because of the productions $L \to \Lambda$, $DR \to R$, $TR \to R$, and $R \to \Lambda$. In order to make it context-sensitive, the markers can be turned into subscripts just as we have done in Exercise 8.23 for the grammar from Example 8.11.

8.28 The proof of Theorem 4.9 provides constructions showing that the family of context-free languages is closed under union, concatenation, and Kleene *. Given context-free grammars G_1 and G_2 one may assume that they do not share any variables. S_1 is the axiom of G_1 and S_2 that of G_2.
$\bigcup$: Combining the productions and adding a new axiom S with productions $S \to S_1$ and $S \to S_2$ yields $L(G_1) \cup L(G_2)$.
$\cdot$: Combining the productions and adding a new axiom S with production $S \to S_1S_2$ yields $L(G_1)L(G_2)$.

$*$: Adding a new axiom S to G_1 with productions $S \to SS_1$ and $S \to \Lambda$ yields $L(G_1)^*$.

The first construction (for union) can also be used for the families of RE languages and of CSLs. Concatenation and Kleene * however need new ideas because of the context-sensitivity in the rewriting process.

Concatenation: Let G_1 be the grammar with production $S_1 \to a$ and let G_2 be the grammar with productions $S_2 \to aB$, $aB \to Ba$, $b \to b$. Thus $L(G_1) = \{a\}$ and $L(G_2) = \{ab, ba\}$.
Using the above construction we obtain $S \Rightarrow S_1S_2 \Rightarrow^2 aaB \Rightarrow aBa \Rightarrow Baa \Rightarrow baa$ which is not in $L(G_1)L(G_2) = \{aab, aba\}$.

Kleene * (for CSLs Kleene + with productions $S \to SS_1 \mid S_1$):
Let G_1 be the context-sensitive grammar given by $S_1 \to BaB$, $aB \to Ba$, $b \to b$. Then $L(G_1) = \{bab, bba\}$.
Using the above construction we obtain $S \Rightarrow^* S_1S_1 \Rightarrow^2 BaBBaB \Rightarrow^4 BBBBaa \Rightarrow^4 bbbbaa$ which is not in $L(G_1)^+$.

8.35 = proof of Theorem 8.25...
Let A be a countable set and let $B \subseteq A$. We must show that B is countable.
If B is finite, there is nothing to prove (B is countable by definition).
Thus assume that B is infinite. Then also A is infinite and since it is countable, there is a bijection f from $\mathbb{N}$ to A. Thus $A = \{f(0), f(1), f(2), \ldots\}$.
Let i_0 be the smallest $i \in \mathbb{N}$ such that $f(i) \in B$ and let, for each $j \geq 1$, i_j be the smallest $i \in \mathbb{N}$ such that $i > i_{j-1}$ and $f(i) \in B$. Thus $B = \{f(i_0), f(i_1), f(i_2), \ldots\}$ and it follows that B is countable since the function g from $\mathbb{N}$ to B defined $g(j) = f(i_j)$ for all $j \in \mathbb{N}$ is a bijection.

8.36 We must show that S is an infinite set, if and only if there is a bijection from S to a proper subset of S.

First assume that S is finite. We have to show that there is no proper subset of S to which a bijection exists from S.
If S is empty, it has no proper subset and we are done. If S consists of only one element, then its only proper subset is the empty set and obviously there exists no bijection from S to the empty set.
We proceed by an inductive argument and consider now a set S with $n+1$ elements for some $n \geq 1$ with as induction hypothesis that no set of n elements allows a bijection to one of its proper subsets.
Now suppose that f is a bijection from S to T, a proper subset of S. Let $s \in S - T$ and g be the restriction of f to $S - \{s\}$. Since f is injective on S,

so is g on $S - \{s\}$. We have that $g(S - \{s\}) = f(S) - \{f(s)\}$ because f is injective. Consequently, $g(S - \{s\})$ is a proper subset of T and hence also a proper subset of $S - \{s\}$, since $s \notin T$.
Thus g is a bijection from $S - \{s\}$ to a proper subset of $S - \{s\}$, contradicting the induction hypothesis.

Conversely, consider an infinite set S. By Theorem 8.25, S has a countably infinite subset $I = \{f(0), f(1), f(2), \ldots\}$ where f is a bijection from $\mathbb{N}$ to I. Let $g : S \to S$ be the function defined by $g(s) = s$ if $s \in S - I$ and $g(f(i)) = f(i+1)$ for all $i \in \mathbb{N}$. This g is a bijection from S to $S - \{f(0)\}$, a proper subset of S.

8.37 Both countability and uncountability are preserved under bijections:
Let $f : S \to T$ be a bijection.
First consider the case that S is countable. If S is finite, then T is finite. If S is countably infinite, then there is a bijection $g : \mathbb{N} \to S$. Then $f \circ g : \mathbb{N} \to T$ is also a bijection, which implies that T is countably infinte.
Next consider the case that T is countable. If T is finite, then S is finite. If T is countably infinite, then there exists a bijection $g : \mathbb{N} \to T$. Observe that $f^{-1} : T \to S$ is a bijection. Consequently, $f^{-1} \circ g : \mathbb{N} \to S$ is a bijection, which implies that S is countably infinite.

8.38 Let S and T be two sets such that S is uncountable and T is countable. Consider $S - T$. Observe that $S = (S - T) \cup (S \cap T)$. Since $S \cap T \subseteq T$, we know from Theorem 8.25 that $S \cap T$ is countable. If $S - T$ would also be countable, then, by Example 8.28, their union $S = (S - T) \cup (S \cap T)$ is countable, a contradiction. Hence it must be the case that $S - T$ is uncountable.

8.39 $\mathbb{Q}$ is countable:
First observe that once we have a bijection g from $\mathbb{N}$ to the nonnegative rational numbers, then we also have a bijection f from $\mathbb{N}$ to $\mathbb{Q}$. Namely, we let $f(0) = g(0)$, $f(2k-1) = g(k)$, and $f(2k) = -g(k)$ for all $k \geq 1$.
We define a bijection from $\mathbb{N}$ to the nonnegative rational numbers by first listing 0 and next the rational numbers represented by pairs of positive integers in "canonical" order (grouped according to the increasing sum of the elements), but leaving out those which have a greatest common divisor larger than 1 (cf. Exercise 8.6(a)), thus guaranteeing that each rational number occurs exactly once:
0, (1,1), (1,2), (2,1), (1,3), (3,1), (1,4), (2,3), (3,2), (4,1), (1,5), (5,1), (1,6), (2,5), (3,4), (4,3), (5,2), (6,1), (1,7), ...

This is similar to the walk through the infinite matrix in Figure 8.27, but double occurrences of rationals — like on the diagonal once we have $(1,1)$, or $(2,4)$ once we have $(1,2)$ — are now avoided.

8.40
a $\mathcal{S}$ is the set consisting of infinite sequences over $\{0,1\}$. Thus each element $s \in \mathcal{S}$ is a function from $\mathbb{N}$ to $\{0,1\}$ giving the symbol (0 or 1) for each position i of $\mathcal{S}$.
$\mathcal{S}$ is uncountable using a direct (diagonal) argument:
Suppose, to the contrary, that $\mathcal{S}$ can be listed as $\mathcal{S} = \{s_0, s_1, s_2, \ldots\}$. Define the sequence/function $s : \mathbb{N} \to \{0,1\}$ by $s(i) = 0$ if $s_i(i) = 1$ and $s(i) = 1$ if $s_i(i) = 0$. Then $s \in \mathcal{S}$, but s does not occur in the list $s_0, s_1, s_2, \ldots$. A contradiction with the asumption that $\mathcal{S} = \{s_0, s_1, s_2, \ldots\}$.

8.41 Determine whether the given set is countable or uncountable.
a The set of all sets $\{a, b, c\}$ consisting of three distinct elements from $\mathbb{N}$ is countable: this follows from **b**, see there.

b The set $\mathcal{F}$ of all *finite* subsets of N is countable. (Contrast this with $2^{\mathbb{N}}$, the set of all subsets of $\mathbb{N}$, which is uncountable by Example 8.31.)
Let, for each $i \in \mathbb{N}$, $\mathcal{F}_i = 2^{\{0,1,\ldots,i\}}$ be the set consisting of all subsets of $\{0, 1, \ldots, i\}$. Thus each $\mathcal{F}_i$ is finite (it has 2^{i+1} elements). Moreover, for every finite subset T of $\mathbb{N}$, there is an i such that $T \in \mathcal{F}_i$. For example, if k is the largest element of T, then $T \subseteq \{0, 1, \ldots, k\}$ which implies that $T \in 2^{\{0,1,\ldots,k\}} = \mathcal{F}_k$.
Hence, $\mathcal{F} = \bigcup_{i=0}^{\infty} \mathcal{F}_i$ and thus a countable union of countable sets, which by Example 8.28 implies that $\mathcal{F}$ is countable.
Now also **a** follows: the set consisting of all three-element subsets of $\mathbb{N}$ is a subset of the countable set $\mathcal{F}$ and therefore countable (by Theorem 8.25).

c The set P of all partitions of $\mathbb{N}$ into a finite number of subsets is uncountable. A partition of $\mathbb{N}$ consists of a finite number of mutually disjoint subsets of $\mathbb{N}$ which together form $\mathbb{N}$. We assume that each subset in the partition must be nonempty (otherwise, there would be even more partitions possible!)
We would like to prove that P is uncountable by establishing a bijection from the set of all subsets of $\mathbb{N}$ to a subset of P: with each set T we would associate the pair $\{T, \mathbb{N} - T\}$. This will not work however, because T or its complement may be empty. Moreover the mapping will not be injective, because it will yield the same pair for T and for its complement. We therefore slightly modify this approach:

Let $V = \{T \subseteq \mathbb{N} \mid T \neq \emptyset \text{ and } 0 \notin T\}$ consist of the nonempty subsets of $\mathbb{N}$ not containing 0. Note that V is uncountable. (The function from $2^{\mathbb{N}}$ to V which maps $\emptyset$ to $\{1\}$ and all other subsets $S \subseteq \mathbb{N}$ to $\{s + 2 \mid s \in S\}$ is injective; thus V has an uncountable subset and must therefore, by Theorem 8.25, itself be uncountable.)
For $T \in V$, define $g(T) = \{T, \mathbb{N} - T\}$. Note that $0 \in \mathbb{N} - T$. It is easy to see that g is injective. Moreover it is surjective on the set $P2$ of all partitions of $\mathbb{N}$ consisting of two sets. Since V is uncountable, it follows that $P2$ is uncountable. Finally, since $P2$ is a subset of the set of all partitions of $\mathbb{N}$ into a finite number of subsets, it follows from Theorem 8.25, that also P is uncountable.

d Since the functions from N to $\{0, 1\}$ correspond one-to-one with the infinite sequences over $\{0, 1\}$, the set of all functions from N to $\{0, 1\}$ is uncountable by Exercise 8.40(a).

e The set of all functions from $\{0, 1\}$ to $\mathbb{N}$ is countable:
There is a one-to-one correspondence between functions $f : \{0, 1\} \to \mathbb{N}$ and pairs $(f(0), f(1)) \in \mathbb{N} \times \mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$ is countable.

f The set of all functions from N to N contains the set from **d** as subset and is therefore uncountable (Theorem 8.25).

g The set of all nondecreasing functions from $\mathbb{N}$ to $\mathbb{N}$ is uncountable:
Consider the following mapping g from the set of all functions from $\mathbb{N}$ to $\mathbb{N}$ (part **f**) to the set of all nondecreasing functions. Let f be an arbitrary function from $\mathbb{N}$ to $\mathbb{N}$. Then the function g_f is defined by $g_f(k) = \sum_{i=0}^{k} f(i)$. It is easily verified that g_f is indeed a nondecreasing function, and that g is indeed a bijection.

h The set of all regular languages over $\{0, 1\}$ is countable according to Theorem 8.25 and Example 8.30: it is a subset of the set of recursively enumerable languages over $\{0, 1\}$, which is countable.

i The set of all context-free languages over $\{0, 1\}$ is countable as in **h**: it is a subset of the set of recursively enumerable languages over $\{0, 1\}$, which is countable.

8.42 $2^{\mathbb{N}}$ is not countable. Give a set $S \subseteq 2^{\mathbb{N}}$ such that both S and $2^{\mathbb{N}} - S$ are uncountable.
Let $S = \{A \subseteq \mathbb{N} \mid A \text{ consists of even integers only}\}$. This set is uncountable since there exists a bijection from S to $2^{\mathbb{N}}$: the function f defined by $f(A) = \{n \mid 2n \in A\}$. Thus S is not countable.
Now consider $2^{\mathbb{N}} - S = \{A \subseteq \mathbb{N} \mid A \text{ contains at least one odd integer}\}$. This set has as a subset $S' = \{A \subseteq \mathbb{N} \mid A \neq \emptyset \text{ and consists of odd integers only}\}$. S' is not countable as follows from the bijection g from S' to the uncountable set $2^{\mathbb{N}} - \{\emptyset\}$ defined by $g(A) = \{n \mid 2n + 1 \in A\}$. Since a countable set has only countable subsets (Theorem 8.25), it must be the case that $2^{\mathbb{N}} - S$ is not countable.

8.43 Let us use $\mathcal{L}_{RE}$ to denote the set of all recursively enumerable languages. Show that the set of languages

$$\mathcal{L} = \{L \subseteq \{0,1\}^* \mid L \notin \mathcal{L}_{RE} \text{ and } \{0,1\}^* - L \notin \mathcal{L}_{RE}\}$$

is uncountable.
By Example 8.30, the set

$$\mathcal{K}_1 = \{L \subseteq \{0,1\}^* \mid L \in \mathcal{L}_{RE}\}$$

of recursively enumerable languages over the alphabet $\{0,1\}$ is countable. Since each language over $\{0,1\}$ is bijectively related to its complement in $\{0,1\}^*$, also the set

$$\mathcal{K}_2 = \{L \subseteq \{0,1\}^* \mid \{0,1\}^* - L \in \mathcal{L}_{RE}\}$$

is countable.
Hence their union

$$\mathcal{K}_1 \cup \mathcal{K}_2 = \{L \subseteq \{0,1\}^* \mid L \in \mathcal{L}_{RE} \text{ or } \{0,1\}^* - L \in \mathcal{L}_{RE}\}$$

is countable (see Example 8.28).
Note that $\mathcal{K}_1 \cup \mathcal{K}_2$ is the complement of $\mathcal{L}$ in the set $2^{\{0,1\}^*}$ of all languages over $\{0,1\}$. As follows from Example 8.31 (as explained right before this example), $2^{\{0,1\}^*}$ is an uncountable set. Since $\mathcal{K}_1 \cup \mathcal{K}_2$ is countable and $2^{\{0,1\}^*} = (\mathcal{K}_1 \cup \mathcal{K}_2) \cup \mathcal{L}$, Example 8.28 implies that $\mathcal{L}$ is uncountable.
