# Fundamentele Informatica 3
# Antwoorden op geselecteerde opgaven uit Hoofdstuk 10

John Martin: Introduction to Languages and the Theory of Computation

Jetty Kleijn

Najaar 2008

**10.6** 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 $L \in \mathcal{L}_{RE}$ 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 exercise 9.12). 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.

**10.8** 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.
HOWEVER, Fermat's last theorem, proved by Andrew Wiles in 1995, says that the set consists only of the elements 1 and 2: we have $1^1 + 2^1 = 3^1$ and $3^2 + 4^2 = 9 + 16 = 25 = 5^2$ for instance, and for no other $n$ there exist integers $x, y, z$ such that $x^n + y^n = z^n$. Hence, given this theorem, we could simply write 1,2 and be done.

**10.10** Given is the computable function $f : \mathbb{N} \to \mathbb{N}$ which is strictly increasing: if $i < j$ then $f(i) < f(j)$. Show that the *range* of $f$ is recursive, where the range of $f$ is $\{m \in \mathbb{N} \mid \exists n \text{ such that } f(n) = m\}$.
From the description of $f$ we conclude that $f$ is a total function. Given an integer $m$, we can determine whether $m$ is in the range of $f$ by computing $f(0), f(1), f(2), \ldots$ until we find an $n$ for which $f(n) \geq m$. If $f(n) > m$, for the first such $n$, then $m$ is not in the range of $f$; otherwise $f(n) = m$ and $m$ is in the range.

**10.11, 10.18**
In 10.11 unrestricted grammars are given through their productions and the exercise is to describe the languages they generate.
In 10.18 CSG's are to be given for these languages.

**a**
$S \to LaR$: the grammar first generates the terminal $a$ inbetween the left-marker $L$ and the right-marker $R$.
$L \to LD \mid \Lambda$, the left-marker either disappears or produces the marker $D$.
$Da \to aaD$ and $DR \to R$: the marker $D$ moves from left to right through the string on the way doubling every occurrence of $a$ it passes; when it hits $R$ it disappears.
$R \to \Lambda$: the right-marker may be erased at any time.
Example derivations:
$S \Rightarrow LaR \Rightarrow aR \Rightarrow a$
$S \Rightarrow LaR \Rightarrow LDaR \Rightarrow LaaDR \Rightarrow LaaR \Rightarrow aaR \Rightarrow aa$

$S \Rightarrow^* LaaR \Rightarrow LDaaR \Rightarrow LaaDaR \Rightarrow LaaaaR \Rightarrow^* a^4$
Note that $L$ may disappear at any moment without obstructing the derivation and it may also derive $D$'s while there are still other $D$'s present. Each of these $D$'s has to "walk" to the right while doubling the $a$'s until it meets $R$; if $R$ disappears while there is still a $D$ present in the string, the derivation will not terminate successfully anymore.
The language generated is $\{a^{2^n} \mid n \geq 0\}$.

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

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

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

**c** The language generated is $\{w \in \{a, b, c\}^* \mid n_a(w) = n_b(w) = n_c(w)\}$.

The grammar given is monotone: the right-hand sides of the productions are never shorter than their left-hand sides.

**d**
$S \rightarrow LA * R$: one group consisting of one $A$ is introduced, surrounded by the two markers $L$ and $R$.
$A \rightarrow a$: each non-terminal $A$ represents a terminal $a$.
$L \rightarrow LI$: the left-marker $L$ introduces a marker $I$.
$IA \rightarrow AI, \quad I* \rightarrow A * IJ$: the $I$ walks to the right passing $A$'s until it meets a symbol $*$. For each $*$ it meets, an extra $A$ is added to the group of $A$'s just left and a new $J$ is introduced.
$JA \rightarrow AJ, \quad J* \rightarrow *J$, and $JR \rightarrow AR$: each $J$ walks to the right until it hits $R$ where it introduces a new $A$.

$IR \to A * R$: the $I$ follows the $J$'s it has introduced, one for each group of $A$'s it has passed; these $J$'s have created a new group of $A$'s just before $R$ and $I$ adds an extra $A$ followed by $*$ just before $R$.
Thus in general, given a string $LA^n * \ldots A^n * R$ consisting of $n$ groups of $n$ $A$'s we obtain $LA^{n+1} * \ldots A^{n+1} * R$ consisting of $n+1$ groups of $n+1$ $A$'s.
$LA \to EA$, $EA \to AE$, $E* \to E$, and $ER \to \Lambda$: are used for termination, the left-marker $L$ changes into the end-marker $E$ which walks from left to right through the string erasing $*$'s on the way until it meets $R$ and both disappear.
Example derivations:
$S \Rightarrow LA * R \Rightarrow EA * R \Rightarrow AE * R \Rightarrow AER \Rightarrow A \Rightarrow a$.
$S \Rightarrow LA * R \Rightarrow LIA * R \Rightarrow LAI * R \Rightarrow LAA * IJR \Rightarrow LAA * IAR \Rightarrow LAA * AIR \Rightarrow LAA * AA * R \Rightarrow EAA * AA * R \Rightarrow^* AAAAER \Rightarrow AAAA \Rightarrow^* a^4$.
$S \Rightarrow^* LAA * AA * R \Rightarrow^* LAAA * AAA * AAA * R \Rightarrow^* a^9$.
The language generated is $\{a^{n^2} \mid n \geq 1\}$.
Note that during a successful derivation a string may contain several occurrences of $I$ and $J$, but each $I$ will stay behind the $J$'s it has itself introduced. $I$'s and $J$'s cannot "overtake" one another.

The grammar above is not context-sensitive (monotone) because of the productions: $E* \to E$ and $ER \to \Lambda$. Following the overall idea of the given grammar we now provide an equivalent monotone grammar. We use subscripts to other symbols present in the string to represent $*$'s, $L$ and $R$. We will have $A_*$ to represent the last $A$ in a group. However, for the last $A$ of the last group the subscripted symbol $A_R$ is used. $A_L$ represents the very first $A$. Moreover all "ordinary" $A$'s are already given as terminal $a$'s. The non-terminals $I$ and $J$ are more or less as before, but they now immediately represent an $a$ (rather than just a position in the string).
$S \to a$ takes care of the shortest word.
$S \to A_L A_* a A_R$ for two groups of two $A$'s (in the old grammar $LAA{*}AA{*}R$).
$A_L \to A_L I$ and $Ia \to aI$: for $L \to LI$ and $IA \to AI$.
$IA_* \to aA_*JI$: for $I* \to A * IJ$. Note that now the $J$ is to the left of the $I$. It will follow $I$ through the string. This is done because $A_R$ is part of the last group of $A$'s and $I$ must now first (before its $J$'s start arriving) add an $a$ to this group and at the same time change $A_R$ to $A_*$ in order to create a place for a new group after this new $A_*$. Consequently we also have the new production
$IA_R \to aA_*aA_R$: for $IR \to A * R$.
$Ja \to aJ$, $JA_* \to A_*J$, and $JA_R \to aA_R$:
for $JA \to AJ$, $J* \to *J$, and $JR \to AR$.

$A_L \to E$, $\quad Ea \to aE$, $\quad EA_* \to aE$, and $EA_R \to aa$:
for $LA \to EA$, $\quad EA \to AE$, $\quad E* \to E$, and $ER \to \Lambda$; note that in this way the very first $A$ walks as $E$ to the right!

**10.12 a** Every derivation begins with the step $S \Rightarrow TD_1D_2$ after which either each of the three symbols is rewritten into $\Lambda$ (hence $\Lambda$ is in the language) or the production $T \to ABCT$ is applied. This production introduces $A$'s, $B$'s, and $C$'s in equal numbers. The two productions $AB \to BA$ and $BA \to AB$ allow one to rearrange the $A$'s and $B$'s in any order, while the productions $CA \to AC$ and $CB \to BC$ 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 $CD_1 \to D_1C$ on the way changing $B$'s into $b$'s with $BD_1 \to D_1b$. 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 $CD_2 \to D_2a$. 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 $BD_1 \to D_1b$ by $B \to b$. Then the $B$ can always terminate, regardless of the positions of the $A$'s and $a$'s.

**10.13 a** To start with we will have productions $S \to ABCDS \mid ABCD$ to generate strings with 4 types of positions in equal numbers;
$BA \to AB$, $CA \to AC$, $DA \to AD$, to move $A$'s to the left;
$CB \to BC$, $DB \to BD$ to move $B$'s past $C$'s and $D$'s to the left;
$DC \to CD$ 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 $S_1 \to A_1BCDS \mid A_1BCD \mid \Lambda$, to have a starter-$A_1$ and to generate $\Lambda$.
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: $A_1A \to A_1A_1$, $A_1B \to A_1B_1$,
from $B$'s to $C$'s: $B_1B \to B_1B_1$, $B_1C \to B_1C_1$,
and from $C$'s to $D$'s: $C_1C \to C_1C_1$, $C_1D \to C_1D_1$, and $D_1D \to D_1D_1$.

Finally, we include the terminating productions $A_1 \to a$, $B_1 \to b$, $C_1 \to a$ and $D_1 \to b$.

Note that the grammar is monotone, except for the production $S_1 \to \Lambda$ which can anyway be omitted if we do not care about having $\Lambda$ 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: $L \to LaA \,|\, LbB$;
it 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: $AM \to MaA'$, $BM \to MbB'$;
it continues as $A'$ or $B'$ respectively: $A'a \to aA'$; $A'b \to bA'$; $B'a \to aB'$; $B'b \to bB'$;
to $R$, at the beginning of the third copy where it leaves its message and then disappears: $A'R \to Ra$, $B'R \to Rb$.

**d** As in **b** 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: $AM \to MA$, $BM \to MB$, $AR \to aRa$, $BN \to bRb$.

**10.17** In Example 10.2 we have seen a grammar generating $\{ss \mid s \in \{a,b\}^*$. It has two productions ($F \to \Lambda$ and $M \to \Lambda$) violating the condition for context-sensitiveness. In this exercise we do not need $\Lambda$ as an element of the language to be generated and we can thus proceed as follows.
$F$ 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 $M$ are used to mark the first symbol in the right half of the word. We thus obtain the following context-sensitive grammar:
$S \to A_1A_2 \,|\, B_1B_2$;

$A_1 \to a$, $B_1 \to b$, $A_2 \to a$, $B_2 \to b$;
alternatively, the first marker symbol can produce an additional $a$ or $b$, and send the corresponding message:
$A_1 \to A_1aA \,|\, A_1bB$, $B_1 \to B_1aA \,|\, B_1bB$;
$Aa \to aA$, $Ab \to bA$, $Ba \to aB$, $Bb \to bB$;
when the messenger arrives at the second marker, this marker produces the terminal:
$AA_2 \to A_2a$, $BA_2 \to A_2b$, $AB_2 \to B_2a$, $BB_2 \to B_2b$.

**10.20** The proof of Theorem 6.1 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 $S \to S_1S$ 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 $S \to S_1S \mid S_1$):
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 $S \Rightarrow^* S_1S_1 \Rightarrow^2 BaBBaB \Rightarrow^* BBBaa \Rightarrow bbbaa$ which is not in $L(G_1)^+$.

**10.24** = proof Lemma 10.2
Let $S$ be a countable set and let $T \subseteq S$. Then $T$ is countable.
If $T$ is finite, there is nothing to prove ($T$ is countable by definition).
Thus assume that $T$ is infinite. Then also $S$ is infinite and since it is countable, there is a bijection $f$ from $\mathbb{N}$ to $S$. Thus $S = \{f(0), f(1), f(2), \ldots\}$.
Let $i_0$ be the smallest $i \in \mathbb{N}$ such that $f(i) \in T$ and let, for each $j \geq 1$, $i_j$ be the smallest $i \in \mathbb{N}$ such that $i > i_{j-1}$ and $f(i) \in T$. Thus $T =$

$\{f(i_0), f(i_1), f(i_2), \ldots\}$ and it follows that $T$ is countable since the function $g$ from $\mathbb{N}$ to $T$ defined $g(j) = f(i_j)$ for all $j \in \mathbb{N}$ is a bijection.

**10.25** $S$ 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 Lemma 10.1, $S$ 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 $S$.

**10.26** 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. Thus 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 countable.

**10.27** 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 Lemma 10.2 that $S \cap T$ is countable. If $S - T$ would also be countable, then, by Theorem 10.13, their union $S = (S - T) \cup (S \cap T)$

is countable, a contradiction. Hence it must be the case that $S - T$ is uncountable.

**10.28** $\mathbb{Q}$ 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 10.8a), 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 10.4, 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.

**10.29** We follow the Convention mentioned in Chapter 9 (page 348), that there are two fixed infinite sets $\mathcal{Q} = \{q_1, q_2, \ldots\}$ and $\mathcal{S} = \{a_1, a_2, \ldots\}$ such that for every Turing machine $T = (Q, \Sigma, \Gamma, q_0, \delta)$, we have $Q \subseteq \mathcal{Q}$ and $\Gamma \subseteq \mathcal{S}$.
Let for each pair $(n,m)$ with $n, m \geq 0$, $\mathcal{T}_{n,m}$ be the set of Turingmachines with state set in $\{q_1, \ldots q_n\}$ and tape alphabet in $\{a_1, \ldots a_m\}$. Each such set is finite and thus countable. Since $\mathbb{N} \times \mathbb{N}$ is countable by Example 10.6, it follows from Theorem 10.13 that the set of all Turing machines $\mathcal{T} = \bigcup_{(n,m) \in \mathbb{N} \times \mathbb{N}} \mathcal{T}_{n,m}$ is countable.

**10.30** $\mathcal{S}$ 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 $\mathcal{S}$.
**a** With each $s \in \mathcal{S}$ we associate the subset of $\mathbb{N}$ which has $s$ as its characteristic funtion. So, define $f : \mathcal{S} \to 2^{\mathbb{N}}$ by $i \in f(s)$ if and only if $s(i) = 1$.
This $f$ is a bijection:
it clearly is a function;
$f$ is injective: assume that $s \neq s'$. Then there is an $i$ such that $s(i) \neq s'(i)$. Consequently, $s(i) = 1$ if and only if $s'(i) = 0$ and so $i \in f(s)$ if and only if $i \notin f(s')$. So $f(s)$ and $f(s')$ differ w.r.t. $i$ and are not the same.
$f$ is surjective: Consider $T \subseteq \mathbb{N}$. Then $T = f(s)$ with s the characteristic

function of $T$: $s(i) = 1$ if $i \in T$ and $s(i) = 0$ otherwise.
It now follows immediately that $\mathcal{S}$ is uncountable: if it would be countable then we would have a bijection $g : \mathbb{N} \to \mathcal{S}$ which — together with the bijection $f$ just defined — forms a bijection $f \circ g : \mathbb{N} \to 2^{\mathbb{N}}$ implying that the latter set is countable, in contradiction with Theorem 10.15.

**b** $\mathcal{S}$ is uncountable using a direct (diagonalization) argument:
Suppose, to the contrary, that $\mathcal{S}$ can be listed as $\mathcal{S} = \{s_0, s_1, s_2, \ldots\}$. Define $s : \mathbb{N} \to \{0, 1\}$ 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 $\mathcal{S} = \{s_0, s_1, s_2, \ldots\}$.

**10.31** 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 Theorem 10.15.)
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 Theorem 10.13 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 Lemma 10.2).

**c** The set $P$ of all finite partitions of $\mathbb{N}$ is uncountable. A partition of $\mathbb{N}$ consists of a finite number of nonempty, mutually disjoint subsets of $\mathbb{N}$ which together form $\mathbb{N}$.
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 $V = \{T \in \mathbb{N} \mid T \neq \emptyset \text{ and } 0 \notin T\}$ consist of the nonempty subsets of $\mathbb{N}$ not containing 0. Note that $2^V$ is uncountable. (The function which maps $\emptyset$ to $\{1\}$ and all other subsets $S \subseteq \mathbb{N}$ to $\{s + 1 \mid s \in S\}$ is injective; thus $2^V$ has an uncountable subset and must therefore, by Lemma 10.2, 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 $2^V$ is uncountable, it follows that $P2$ is uncountable. Finally, since $P2$ is a subset of the set of all finite partitions of $\mathbb{N}$ it follows from Lemma 10.2, 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 10.30.

**e** The set of all functions from $\{0, 1\}$ to $\mathbb{N}$ is countable:
There is a one-to-one correspondence between functions $f : \mathbb{N} \to \{0, 1\}$ 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 (Lemma 10.2).

**g** The set of all nonincreasing functions from $\mathbb{N}$ to $\mathbb{N}$ is countable:
$f : \mathbb{N} \to \mathbb{N}$ is nonincreasing if for all $i \leq j$ we have $f(j) \leq f(i)$. Therefore, if $f$ is nonincreasing all values of $f$ are bounded by $f(0)$. Consequently, the range of $f$ is a finite subset $\{a_0, a_1, a_2, \ldots a_k\}$ of $\mathbb{N}$ with $a_0 > a_1 > a_2 > \ldots > a_k$. Thus $f$ is fully specified once also the points $n_1 < n_2 < \ldots < n_k$ in $\mathbb{N}$ are given where the value of $f$ "drops": $f(0) = \ldots = f(n_1 - 1) = a_0$, $f(n_1) = \ldots = f(n_2 - 1) = a_1$, ... , $f(n_{k-1}) = \ldots = f(n_k - 1) = a_{k-1}$, $f(m) = a_k$ for all $m \geq n_k$.
Now $f$ is completely determined by the pair $(\{a_0, a_1, \ldots, a_k\}, \{n_1, \ldots, n_k\})$. Note that if $k = 0$, then $f$ is the constant function $f(x) = a_0$ for all $x \in \mathbb{N}$ corresponding to $(\{a_0\}, \emptyset)$.
All this shows that $\mathcal{S}$, the set of all nonincreasing functions from $\mathbb{N}$ to $\mathbb{N}$, can be seen as a subset of $\mathcal{F} \times \mathcal{F}$ where $\mathcal{F}$ is the set of all finite subsets of $\mathbb{N}$ which was shown to be countable in **b**. As in Example 10.6, $\mathcal{F} \times \mathcal{F}$ is countable. Finally, by Lemma 10.2, it follows that $\mathcal{S}$ is countable.

**h** The set of all regular languages over $\{0, 1\}$ is countable according to Lemma 10.2 and Example 10.8: it is a subset of the set of recursively enumerable languages 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 which is countable.

**10.32** $2^{\mathbb{N}}$ 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 $2^{\mathbb{N}} - S = \{A \subseteq \mathbb{N} \mid A$ contains at least one odd integer$\}$. This set has as a subset $S' = \{A \subseteq \mathbb{N} \mid A \neq \emptyset$ 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 (Lemma 10.2), it must be the case that $2^{\mathbb{N}} - S$ is not countable.

**10.33** 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.
$\mathcal{L}_{RE}$ is countable by Example 10.8. Consequently, by Lemma 10.2, 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 Theorem 10.13).
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 $\{0,1\}$. By Theorem 10.15, $2^{\{0,1\}^*}$ is an uncountable set. Since $\mathcal{K}_1 \cup \mathcal{K}_2$ is countable and $2^{\{0,1\}^*} = (\mathcal{K}_1 \cup \mathcal{K}_2) \cup \mathcal{L}$, Theorem 10.13 implies that $\mathcal{L}$ is uncountable.

---

versie 06 januari 2009