Assignment G6

Morten Fausing & Jens Kristian Refsgaard Nielsen & Thomas Vinther

27. april 2019

1 Countable and Uncountable Sets

Following exercise 8.42 in the book: We know that $2^{\mathbb{N}}$ is uncountable. Of course its complement, $\emptyset$ is countable. Question: Define a subset $S \subseteq 2^{\mathbb{N}}$, such that both S and its complement $S^c := 2^{\mathbb{N}} \setminus S$ are uncountable.
Here follows a sneaky solution.
We know that $\mathbb{R}$ is uncountable, so there exists a bijection $b : \mathbb{R} \to 2^{\mathbb{N}}$, we also know that all non empty open real intervals are uncountable, for instance $(0,1) \subset \mathbb{R}$, as we can map $\mathbb{R}$ bijectively onto $(0,1)$ via a bijection $r : \mathbb{R} \to (0,1)$. Consider the commutative diagram

$$\begin{tikzcd} 2^{\mathbb{N}} \arrow[r, "b^{-1}"] & \mathbb{R} \arrow[d, "r"] \\ S \arrow[u, "\subseteq"] \arrow[r, "f"] & (0,1) \end{tikzcd}$$

Choose then the set $S = b((0,1)) \subsetneq 2^{\mathbb{N}}$ and then notice how

$$\begin{aligned} b(r^{-1}(b^{-1}(S))) &= b(r^{-1}(b^{-1}(b((0,1))))) = b(r^{-1}((0,1))) \\ &= b(\mathbb{R}) = 2^{\mathbb{N}} \end{aligned}$$

So S stands in bijection to $2^{\mathbb{N}}$ via the bijective composition of bijections

$$f = b \circ r^{-1} \circ b^{-1}|_S : S \to 2^{\mathbb{N}}$$

Where $|_S$ denotes the restriction of domain to S, or equivivalently composition with the inclusion injection. The restriction of a bijection is clearly injective, and above we saw that the function is surjective, so f is a bijection from S to $2^{\mathbb{N}}$. On the other hand $b^{-1}(2^{\mathbb{N}} \setminus S) = \mathbb{R} \setminus (0,1)$ which clearly is uncountable as well, for instance the interval (2,3) is in there and we saw that those are uncountable. $\square$

2 Enumeration and Recursively Enumerable

Assume that TM M_1 enumerates on a tape all words in L "in the weak sense". That is, it is possibly that duplicate words appear on the tape, but you can at least assume that M_1 writes every word from L to the tape at least once (and no other words).
Question: Prove that in this case, L must be recursively enumerable, i.e. there exists a TM M_2 that accepts the same language L.
Proof: We define a machine M that lets M_1 enumerate 1 word on tape 2. It then checks if the word is equal to the input word on tape one. If yes then the machine accepts. If not M_1 enumerates the next word and M performs the equality check. Should M_1 finish the enumeration and fail all equality checks M rejects the input word.
Clearly M accepts all words enumerated by M_1, and thus $\mathcal{L}(M) = L$ since the words in L are exactly those that M_1 enumerates, and M accepts no other words. The possibility of duplicate enumerations are of no consequence to M, other than of course the runtime disadvantage but that is unimportant in this context.

3 Halting Problem Variant

Question: Assuming $L(M_0)$ is not recursive, prove that the language, $L = \{w|M_0 \textit{ halts on input } w\}$ is not recursive
Proof: The assumption that $L(M_0)$ is non recursive is equivalent to saying that $L(M_0)$ is recursive enumerable, and not recursive, i.e. has at least infinitely many inputs that it hangs on. For this task we only need one w that it hangs on. Assume for contradiction that there exists a TM H such that $L = \mathcal{L}(H)$ and H always halts, i.e. that L is recursive.
Now M_0 hangs on some w, so H should reject w because H accepts exactly the inputs that M_0 halts on. But since M_0 hangs on w, H cannot accept or reject w given that it can only make a decision when M_0 has made one. H will therefore hang on w, which is a contradiction, since we assumed that H always halts. $\square$