

1. Prove that the Turing-acceptable languages are closed under union, intersection, and reversal. For each property, give a detailed sketch of the proof, by saying how you would build a Turing machine that accepts the resulting language, given the Turing machine(s) that accept the original language(s).

Union: I would build a Turing machine that utilizes the structure of multiple tapes from the notes. There would be 1 input tape and 1 head tape for each language in the union. The input tape for each language would be the same w . Then, run each machine on the respective tape, updating the head as needed. If any of them halt, then the new Turing machine will accept. Otherwise, it will not accept if none of the machines halt.

Intersection: Like union, I would again build a Turing machine that utilizes the structure of multiple tapes from the notes. There would be 1 input tape and 1 head tape for each language in the intersection. The input tape for each language would be the same w . Then, run each machine on the respective tape, updating the head as needed. If any of them do not halt, then the new Turing machine will not accept. Otherwise, it will accept if all machines halt.

Reversal: For reversal, I would implement the strategy that was used for the 2-way infinite tape. The top layer of the tape would consist of the reversed w . Then, I would have the machine copy the reversed w to the bottom layer to make it non-reversed. To do that, it would work from right-to-left reading the top layer and copy left-to-right on the bottom layer. Finally, using the original Turing machine, I'd run through the bottom layer and if the original machine accepts, the new Turing machine will also accept.

2. Prove or disprove that the Turing-acceptable languages are closed under concatenation.

Turing-acceptable languages are NOT closed under concatenation. I can prove this by defining 2 languages: $L_1 = \{0^n 1^n : n \geq 0\}$ and $L_2 = \{0^n 1^n : n \geq 0\}$. Both of these languages are Turing-acceptable.

Using how up above I proved how Turing-acceptable languages are closed under reversal, that means that L_2 can be reversed, which is $L_{2R} = \{1^n 0^n : n \geq 0\}$, and that is a Turing-acceptable language.

However, when we concatenate 2 Turing-acceptable languages L_1 and L_{2R} , we get the language $L = \{0^n 1^n 1^m 0^m : n, m \geq 0\}$. The machine would have to be able to tell where the string splits, but it cannot definitively know where the split occurs. Without knowing the split, the new machine won't be able to accept the language. Therefore, it won't always be able to accept the 2 Turing-acceptable languages concatenated. Thus, **Turing-acceptable languages are NOT closed under concatenation.**