To prove this problem:

We need to show equivalence between a Turing machine that decides a language and an enumerator that enumerates it. Hence we have to show the proof in both directions i.e.,

1. Language is decidable then enumerator enumerates the language in lexicographic order.

2. Enumerator enumerates the language in lexicographic order and then it is decidable.

1. Language is decidable then enumerator enumerates the language in lexicographic order.

Proof:

If direction:

Let us assume that we have a Turing machine M to decide a language L.

Now we can use this M to construct an enumerator E as follows.

We generate the strings in the lexicographic order and input each string into M for L.

If M accepts then print that string. Therefore E prints all strings of L in lexicographic order.

2. Enumerator enumerates the language in lexicographic order and then it is decidable.

Proof:

And if direction: -

Now we need to consider the other direction.

That is, if we have an enumerator E for a language L, then we can use E to construct a Turing machine M that decides L.

Here we have to consider two cases.

Case (i): If L is finite:

If L is finite language, then it is decidable, because all finite languages are decidable.

Case(ii) If L is infinite:-

If L is infinite then a decider for L operates as follows.

• On receiving the input w, the decider enumerates all strings of L in lexicographic order until a string greater than w appears in the lexicographic order.

• This must eventually occur since L is infinite.

• If w has appeared in the enumeration already, then *accept*.

• If w has not yet appeared in the enumeration then it will never appear and hence we can *reject*.

So in both cases L is decidable. The theorem proved in both directions.

Therefore,

A language is decidable if and only if some enumerator enumerates the language in lexicographic order.