获得的答案 返回

As per the theorem 3.16, Every DTM (Deterministic Turing machine) has an equivalent NDTM (Non-deterministic Turing machine). The corollary 3.19, states that a language is decidable iff some NDTM (Non-deterministic Turing machine) decides it.

In order to modify the proof of theorem 3.16 to obtain corollary 3.19, prove the following:

(i) If a Language *L* is decidable then the language *L* can be decided by NDTM (non-deterministic Turing machine).

Let *L* be a language. The Language *L* is decidable if it is decided by a deterministic Turing machine. Any DTM (Deterministic Turing machine) is automatically a NDTM (Non-deterministic Turing machine).

Thus, language L is decidable if it is decided by NDTM (Non-deterministic Turing machine) TM also.

(ii) If some NDTM (Non-deterministic Turing machine) decides a language then it is decidable.

Let *L* be an any language and it is decided by NDTM (Non-deterministic Turing machine).

Construct a deterministic TM *D* that decides *L* and prove that *D* simulates *N*.

Construction of *D* and Simulation of *D* with *N*:

The simulating deterministic TM '*D*' has 3 tapes

1. Input tape

2. Simulating tape

3. Address tape

By the theorem

"Every multi tape TM has an equivalent single tape TM"

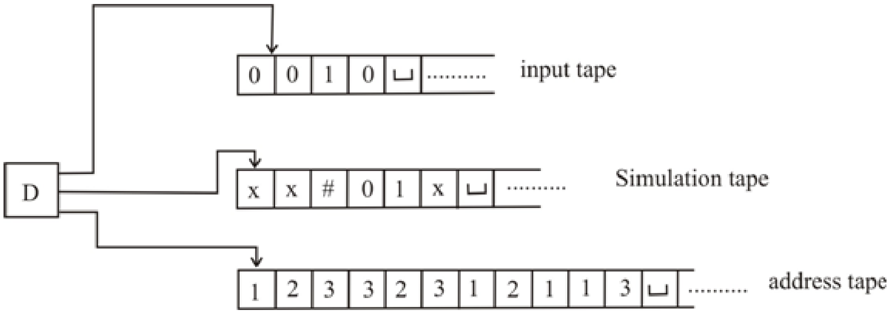
D's arrangement is equivalent to having a single tape TM.

The machine *D* uses its three tapes in the following way:

- Tape 1 always contains the input string and is never altered.

- Tape 2 maintains a copy of *N*'s tape on some branch of its non-deterministic computation

- Tape 3 keeps track of *D*'s location in N's non-deterministic computation tree.



The description of *D* in detail as follows:

Stage 1: Initially tape 1 contains the input *w*, and tapes 2 and 3 are empty.

Stage 2: Copy tape 1 to tape 2.

Stage 3: Use tape 2 to simulate *N* with input *w* on one branch of its non-deterministic computation. Before each step of *N* check the next symbol on tape 3 to determine which choice to make among those allowed by *N*'s transition function. If no more symbols left on tape 3 or if this non-deterministic choice is invalid, abort this branch by going to stage 4. Also go to stage 4 if a rejecting configuration is encountered. If an accepting configuration is encountered, then accept the input.

Stage 4: Replace the string on tape 3 with the lexicographically next string. Simulate the next branch of *N*'s computation by going to stage 2.

Stage 5: Reject if all branches of non-determinism of *N* are exhausted.

Now it can be said that *D* is a decider of *L*.

- If *N* accepts its input, then *D* will find an accepting branch and accept.

- If *N* rejects its input, then all its branches halt and reject.

Thus, the language *L* is decidable.

From (i) and (ii),

The language *L* is decidable if and only if some non-deterministic Turing machine decides it.