A Nondeterministic algorithm for $*$ of a decidable language

Let L be a decidable language accepted by a decider DTM M.

$w \in L^*$ iff

$\exists$ k and $\exists$ a partition of $w = w_1 \dots w_k$ such that

$w_i \in L \; \forall i = 1, 2, \dots k$.

A NTM decider M' accepts L^* as follows:

GUESS the positions $i1, \cdots i_k$ where w is chopped.

(Traverse the input from the left end; at each position GUESS whether or not to mark the symbol.)

Simulate M on each piece of w one after the other.

Accept iff all the pieces are accepted by M.

A Nondeterministic algorithm for compositeness

Input: $W = w_1 w_2 \cdots w_n$, an n-bit number with $W \geq 2$.

Problem: Accept iff W is composite.

A two-tape NTM decides if W is composite as follows. Tape 1 contains the input W.

1. If $W = 2$ then reject;
2. GUESS a number X and write it on tape 2:

   Until the end of the input do:

      Nondeterministically write 0/1 on tape 2; move right by one position on the input tape and tape 2;

3. If $X \leq 1$ reject;
4. Write a separator at the right most end of tape 2;
5. Go to the left most position of the input tape;
6. GUESS a number Y and write on tape 2 after X:

   Until the end of the input do:

      Nondeterministically write 0/1 on tape 2; move right by one position on the input tape and tape 2;

7. If $Y \leq 1$ reject;
8. If $XY = W$ accept else reject;

Path finding in a Graph

Given: a directed graph $G = (V, E)$, where $V = \{1, 2, \cdots, n\}$

Problem: Decide if there is a directed path from vertex 1 to vertex n.

Assume input is the adjacency matrix A for G:

$A[i, j] = 1$ if $(i, j) \in E$, 0 otherwise.

Compute $NameLength = \lceil \log_2 n \rceil$.

Let $CurrentVertex \leftarrow 1$.

Let $EdgesSeen \leftarrow 0$.

Repeat until $EdgesSeen = n$:

> GUESS a vertex $NextVertex$:
>
> > (guess a 0/1 binary number of length $NameLength$.)
>
> If $A[CurrentVertex, NextVertex] = 0$ HALT and REJECT.
>
> If $NextVertex = n$ HALT and ACCEPT.
>
> $CurrentVertex \leftarrow NextVertex$.
>
> Increment $EdgesSeen$.

HALT and REJECT.

Finding a K-Clique in a Graph

Given: a directed graph $G = (V, E)$, where $V = \{1, 2, \cdots, n\}$, and an integer K with $1 \leq K \leq n$.

Problem: Decide if there is a complete graph of size K in G.

Assume graph G is given as an adjacency matrix A:

$A[i, j] = 1$ if $(i, j) \in E$, 0 otherwise.

Compute $NameLength = \lceil \log_2 n \rceil$.

Copy K to Tape 2.

Repeat until Tape 2 is 0:

Write a separator mark # on Tape 3.

GUESS a vertex u and write it on Tape 3.

(guess a 0/1 bit number with length $NameLength$.)

Decrement number in Tape 2.

Repeat until no more vertices in Tape 3:

Read the first vertex from Tape 3 and store it in CurrentVertex.

Delete the first vertex from Tape 3.

For all vertices u in Tape 3 do:

If $A[CurrentVertex, u] = 0$ halt and reject.

Halt and accept.

Finding a K-Clique in a Graph

Given: a directed graph $G = (V, E)$, where $V = \{1, 2, \cdots, n\}$, and an integer K with $1 \leq K \leq n$.

Problem: Decide if there is a complete graph of size K in G.

Assume graph G is given as an adjacency matrix A:

$A[i, j] = 1$ if $(i, j) \in E$, 0 otherwise.

GUESS K vertices $u_1, \ldots, u_K$.

For $1 \leq i \leq K$ do:

  For $1 \leq j \leq K$ with $j \neq i$ do:

    If $A[u_i, u_j] = 0$ halt and reject.

Halt and accept.