Definition 1. *Let $L \subseteq \Sigma^*$. We say $L \in$***NP** *if there is a two-place predicate $R \subseteq \Sigma^* \times \Sigma^*$ such that R is computable in polynomial time, and such that for some $c, d \in \mathbb{N}$ we have for all $x \in \Sigma^*$*
$x \in L \Leftrightarrow$ there exists $y \in \Sigma^*, |y| \leq c|x|^d$ and $R(x, y)$.

1 Nondeterminism

DTM (deterministic TM): next step of computation is completely determined by the current configuration of a TM
NTM (nondeterministic TM): there may be several possibilities for a next step

Formally. NTM's transition function is $\delta : Q\times\Sigma \rightarrow 2^{Q\times\Sigma\times\{L,R,-\}}$, i.e., a pair (state,symbol) is mapped to a set $\{$(state,symbol,movement) $\}$.

Thus, DTM's computation is a path; whereas NTM's computation is a tree.

Accept/Reject criteria for NTMs: $x \in L$ iff there *exists some* accepting computation.

Note: if $x \in L$, then there may be some rejecting computations as well.

Time $= \max_{\text{paths } p} \{\text{length of } p\}$

Space $= \max_{\text{paths } p} \{\text{number of work-tape cells touched on a path } p\}$

2 Nondeterministic Complexity Classes

$\mathsf{NTime}(f(n)) = \{L \mid \text{some multi-tape NTM decides } L \text{ in time at most } f(n)\}$

$\mathsf{NSpace}(f(n)) = \{L \mid \text{some multi-tape NTM decides } L \text{ in space at most } f(n)\}$

- $\mathsf{NP} = \cup_k \mathsf{NTime}(n^k)$
- $\mathsf{NEXP} = \cup_k \mathsf{NTime}(2^{n^k})$
- $\mathsf{NL} = \mathsf{NSpace}(\log n)$
- $\mathsf{NPSPACE} = \cup_k \mathsf{NSpace}(n^k)$

Remark We'll see later that $\mathsf{NPSPACE} = \mathsf{PSPACE}$, and so there is really no need to use the name $\mathsf{NPSPACE}$.

Recall our previous definition of NP: $L \in \mathsf{NP}$ if there is a relation $R \in \mathsf{P}$ and a constant c such that

$$L = \{x \mid \exists y, |y| \leq |x|^c, R(x, y)\}$$

Lemma 2. *The two given definitions of* NP *are equivalent.*

Proof. First we will show that every language computable by nondeterministic TM in polynomial time can be represented in the form $\exists y, |y| \leq |x|^c, R(x,y)$ for some c, d and R. For simplicity, assume that the machine has at most 2 possible choices for every transition (that is, its computation tree is binary).

Let L be a language and M a polytime NTM computing L. Let x be a string over the alphabet of L. Consider a computation of M on x, represented as a tree. By our assumption, this tree is binary. Now, if x is in the language, then there exists a sequence of choices (encodes as a binary string) that M makes to get to an accepting configuration. If x is not in the language, then any sequence of choices ends in a rejecting configuration. Now, take y to be the sequence of choices, and R a check that M can get to accepting configuration by following this sequence of choices. Note that R is deterministic polynomial-time computable, since all nondeterministic choices of M are encoded in y. If M runs in time n^k, we can take $c = 1$, $d = k$, since the number of nondeterministic choices cannot be larger than the running time of M.

For the other direction, consider a language L represented as $\exists y |y| \leq c|x|^d R(x,y)$. Make a complete binary tree in which each path corresponds to a possible y; there will be $2^{c|x|^d}$ such paths. At the end of every path, run a computation of R on x and the y corresponding to this path; this computation is polynomial-time. If at any leaf of the tree R accepted, accept.

□

NTMs cannot be efficiently implemented. So they are an abstraction. But, a huge number of real-life problems are in NP because they are of the form: problem description such that a solution to the problem is "small" and the solution is "easy" to test for correctness. (So we can nondeterministically guess a solution, and then test its correctness in polytime.)

3 Relations

Lemma 3. *$L \subseteq NL$, P $\subseteq$ NP, $PSPACE \subseteq NPSPACE$ and EXP $\subseteq$ NEXP*

Proof. Trivial: a DTM is a special case of an NTM. In fact, PSPACE=NPSPACE, we will show it later. □

Theorem 4. NP $\subseteq$ PSPACE

Proof. Try all possible y's (of polynomial length) sequentially. The only information that needs to be preserved is which y is being used at the moment. □

Theorem 5. NL $\subseteq$ P

Proof. Configuration graph of an NL NTM has only polynomially many configurations, just like for an L TM. Therefore, a path from start to an accepting configuration can be evaluated by a breadth-first or depth-first search, which can be done in polynomial time. □

Major Open Problem: $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$

This is a problem of "Generating a solution vs. Recognizing a solution". Some examples: student vs. grader; composer vs. listener; writer vs. reader; mathematician vs. computer.