CS F364
Design & Analysis of Algorithms

ALGORITHMS - COMPLEXITY

Non-Deterministic Computation

- Ideas, Examples
- Certification Model

NON-DETERMINISTIC COMPUTATIONS - DEFINITION

- In a deterministic TM:
  - Computation proceeds by moving from one state to another in a "deterministic way"
    - i.e. given a state, and an input, the next state can be "determined"
    - i.e. the state transitions can be defined by a function (on set of states and set of symbols)
- By contrast, in a non-deterministic TM:
  - Computation proceeds by moving from one state to one of many states in a "non-deterministic" way
    - i.e. given a state, and an input, the next state is *not known a-priori* (before the transition)
    - i.e. the state transitions must be defined by a relation (on set of states and set of symbols)

NON-DETERMINISTIC COMPUTATIONS - DEFINITION [2]

- Non-deterministic computations are defined using abstract machines:
  - There are no inherently non-deterministic computers (at least as on today – assuming no Schrodinger's cats!)

- Conceptually, a non-deterministic computation proceeds by making "non-deterministic" choices in real-time i.e.
  - Each choice takes O(1) time irrespective of the number of options available.

Non-Deterministic Algorithms - Example 1

- Problem:
  - Given an array A of values, and a key k, find whether a value with key k is contained in A.
  - Algorithm NDSearch(A, k)

```
len = A.length;
ind = choose(0, len-1);
if (A[ind]==k) return 1;
else return 0;
```

Non-Deterministic Algorithms - Example 2

- **Problem SAT(isfiability):**
  - Given a Boolean expression E with variables x_0,..., x_{n-1} find whether there exists an assignment of (Boolean) values that satisfies E.
  - NDSAT(E, X)

```
len = X.length;
for(j=0;j <len;j++)     {  X[j] = choose(0, 1);  }
if (evaluate (E, X) == 1)    return 1;
else return 0;
```

Non-Deterministic Algorithms - Characteristics

- Observations:
  - When does a non-deterministic algorithm work?
    - OR When does it fail?
  - How much time does it take?
    - OR does **Choose** do an exhaustive search?
- Ground Rules:
  - When a non-deterministic algorithm returns 1 it is correct.
  - When it returns 0 it may be incorrect

ND Computations – Alternative Perspective

- A non-deterministic algorithm verifies a "certified solution" if one is presented.
  - Example 1: (NDSearch)
    - …
    - ind = *choose(0, len-1)*; // ind is a certificate
    - if (*A[ind]==k*) return 1 // verification
    - …
  - Example 2: (NDSAT)
    - …
    - for(j=0; j <len; j++)
      X[j] = *choose(0, 1)*; // (values) X is a certificate
    - if (*evaluate (E, X) == 1*) // verification
    - …