

## Complexity Classes

A *program* accepts a language  $L$  in time  $T(n)$  if every  $w \in L$  is accepted in  $O(T(|w|))$  moves.

For *nondeterministic programs*, this means at least one sequence of moves of length  $O(T(n))$  leads to acceptance.

$DTIME(T(n))$  = set of languages accepted by  $T(n)$  deterministic programs

$NTIME(T(n))$  = set of languages accepted by  $T(n)$  nondeterministic programs

Regular languages  $\subseteq DTIME(n)$ .

Each regular language is accepted by a DFA, which reads each symbol once.

Context-free languages  $\subseteq DTIME(n^3)$ .

The CYK algorithm is  $O(n^3)$ .

Context-free languages  $\subseteq NTIME(n)$ .

Each CFL corresponds to a CFG in Chomsky normal form. A derivation of a length  $n$  string is  $O(n)$  long, so  $O(n)$  nondeterministic moves.

$DTIME(f(n)) \subset$  recursive languages provided that  $f(n)$  is computable.

## Complexity of Recursive Languages

Theorem: If  $f(n)$  is computable, then  $DTIME(f(n)) \neq$  recursive languages.

Proof: by diagonalization

Enumerate inputs  $x$  and programs  $M$ .

Let  $\bar{L} = \{x_i : M_i \text{ accepts } x_i \text{ in } n f(n) \text{ steps}\}$

$L$  is recursive because we can generate program  $M_i$  and run it for  $n f(n)$  steps.

---

Assume  $L \in DTIME(f(n))$ .

This implies some  $M_j$  runs in time  $c f(n)$ .

As needed, we add dead code to  $M_j$  to create  $M_k$  such that  $|x_k| > c$ .

However, if  $M_k$  accepts  $x_k$  then  $x_k \notin L$ ,  
and if  $M_k$  rejects  $x_k$ , then  $x_k \in L$ .

This is a contradiction, so  $L \notin DTIME(f(n))$ .

## Other Complexity Results

For every integer  $k \geq 1$ .

$$DTIME(n^k) \subset DTIME(n^{k+1})$$

This implies that there are problems in  $DTIME(n^4)$  that are not in  $DTIME(n^3)$ .