



Limits of Computation

13 - Complexity Classes
Bernhard Reus

1



The complexity story so far

- time measure for machine like languages
- and for WHILE
- discussed fairness
- comparing timed programming languages



Time Complexity

THIS TIME

- We deal with worst case time complexity, so we need
- upper bounds of runtime (as a function)
- Define *complexity classes* for *programs* (*asymptotic time complexity*)
- from which we derive *complexity classes* for *problems*.

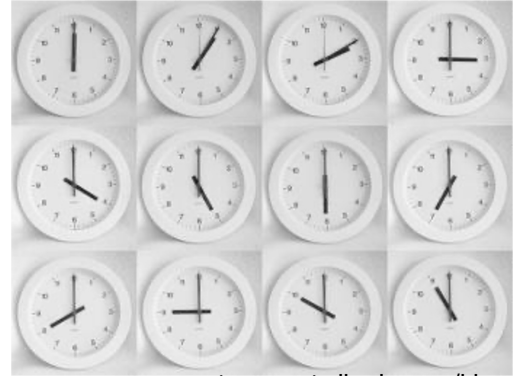


image: mindhacks.com/blog



Runtime Bounds: functions describing worst-case complexity



Runtime Bounds

- A *runtime bound* is a total function on the natural numbers:

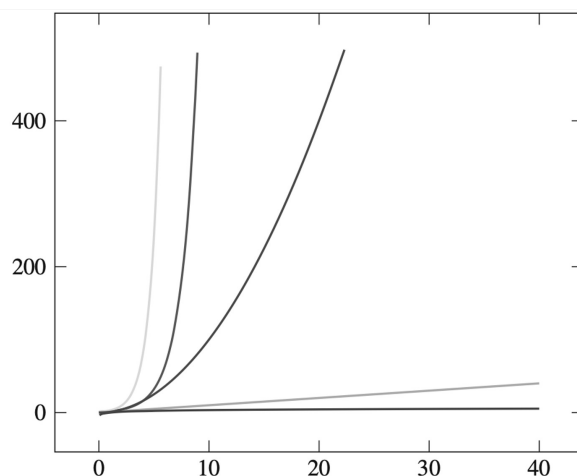
$$f : \mathbb{N} \rightarrow \mathbb{N}$$

- which maps the size of the input of a program to the time it takes **at most** to run the program with the respective input,
- such that for *all* inputs the program's runtime is bounded by the runtime bound (worst case).
- The time bound f is a function on natural numbers, as it must depend on size of program input d .

if d is a binary tree then its size, $|d|$, is the number of leaves, i.e. of nils in the tree



Runtime Bound Examples



Asymptotic time complexity means that we will be mainly interested in the runtime bound's behaviour for large input, i.e. what happens in the limit!

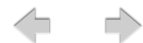
... large size

Graph of $\log_2 x$ (magenta), x (green), x^2 (blue), 2^x (red), 3^x (yellow)



Classifying Programs by Runtime

- Already discussed “robustness” w.r.t computability (Church-Turing thesis)
- now with resource-bounds = time bound for running time
- classify sets of programs that run within the same time bound
- distinguish four classes of programs
 - * those with a given function as time bound,
 - * those with all polynomial, linear, and exponential functions, respectively, as time bounds.



Programs with Bounds

Definition (*programs with bounds*) Given a timed programming language L and a total function $f : \mathbb{N} \rightarrow \mathbb{N}$, we define four classes (sets) of *time bounded programs*:

1. $L^{time(f)} = \{p \in L\text{-programs} \mid time_p^L(d) \leq f(|d|) \text{ for all } d \in L\text{-data}\}$
 This is the set of programs that have a runtime that is bounded by function f .
size of input

2. $L^{ptime} = \bigcup_{p \text{ is a polynomial}} L^{time(p(n))}$
 This set is the union of all set of L programs that have a polynomial function p as time bound. Recall that a polynomial function is of the form

$$c_k \times n^k + c_{k-1} \times n^{k-1} + \dots + c_2 \times n^2 + c_1 \times n + c_0 .$$



Programs with Bounds (II)

3. $L^{ltime} = \bigcup_{k \geq 0} L^{time(k \times n)}$

This set is the union of all sets of L programs that have a linear function f as time bound. *Linear functions* are of the form $f(n) = k \times n$.

4. $L^{exptime} = \bigcup_{p \text{ is a polynomial}} L^{time(2^{p(n)})}$

This set is the union of all set of L programs that have an exponential function f as time bound. An exponential functions are of the form $f(n) = 2^{p(n)}$ where p is polynomial.



From Classes of Programs to Classes of Problems



Fix L-data

- To be able to compare different machine models (notions of computability) we fix the data type of programs (i.e. programming languages):
- $L\text{-data} = \{0,1\}^*$ i.e. words made up of 0 and 1's, e.g. 11011, 00001100. For instance, we can encode $\{0,1\}^*$ in WHILE-data (binary trees), as lists of 0s and 1s;

1101 = [1,1,0,1]
- 'Problems' are sets (properties) of such words.
- The size of such words is the length, e.g. $|1101| = 4$

12



Problem Classes

Definition (*complexity classes*) Given a timed programming language L and a total function $f : \mathbb{N} \rightarrow \mathbb{N}$, we define four (complexity) classes of problems:

1. The class of *problems* L -decidable in time f is:

$$\mathbf{TIME}^L(f) = \{A \subseteq \{0,1\}^* \mid A \text{ is decided by some } p \in L^{time(f)}\}$$

In other words, this is the class of all problems (or sets) A about words over alphabet $\{0,1\}$ that are decided by an L -program with a runtime that is bounded by time bound (function) f .

2. The class \mathbf{P}^L of *problems* L -decidable in polynomial time is:

$$\mathbf{P}^L = \{A \subseteq \{0,1\}^* \mid A \text{ is decided by some } p \in L^{ptime}\}$$

In other words, this is the class of all problems (or sets) A about words over alphabet $\{0,1\}$ that are decided by an L -program with a runtime that is bounded by a polynomial.

14



Problem Classes (II)

3. The class \mathbf{LIN}^L of *problems L-decidable in linear time* is:

$$\mathbf{LIN}^L = \{A \subseteq \{0, 1\}^* \mid A \text{ is decided by some } p \in \mathbb{L}^{ltime}\}$$

In other words, this is the class of all problems (or sets) A about words over alphabet $\{0, 1\}$ that are decided by an L -program with a runtime that is bounded by a linear function.

4. The class \mathbf{EXP}^L of *problems L-decidable in exponential time* is:

$$\mathbf{EXP}^L = \{A \subseteq \{0, 1\}^* \mid A \text{ is decided by some } p \in \mathbb{L}^{exptime}\}$$

In other words, this is the class of all problems (or sets) A about words over alphabet $\{0, 1\}$ that are decided by an L -program with a runtime that is bounded by an exponential function.



Lifting Simulation Properties to Classes

Lemma $\mathbb{L} \preceq^{ltime} \mathbb{M}$ implies $\mathbf{LIN}^L \subseteq \mathbf{LIN}^M$, and as a consequence, $\mathbb{L} \equiv^{ltime} \mathbb{M}$ implies $\mathbf{LIN}^L = \mathbf{LIN}^M$.

This means that if L can be simulated by M up to linear time difference then every problem in \mathbf{LIN}^L is already in \mathbf{LIN}^M . And then, as a consequence, that if L and M are linearly equivalent then \mathbf{LIN}^L and \mathbf{LIN}^M are the same.

Proof in exercises.

Similar results can be shown for \mathbf{P}



END

© 2008-23. Bernhard Reus, University of Sussex

Next time:
Robustness of P