Limits of Computation

13 - Complexity Classes

Bernhard Reus

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

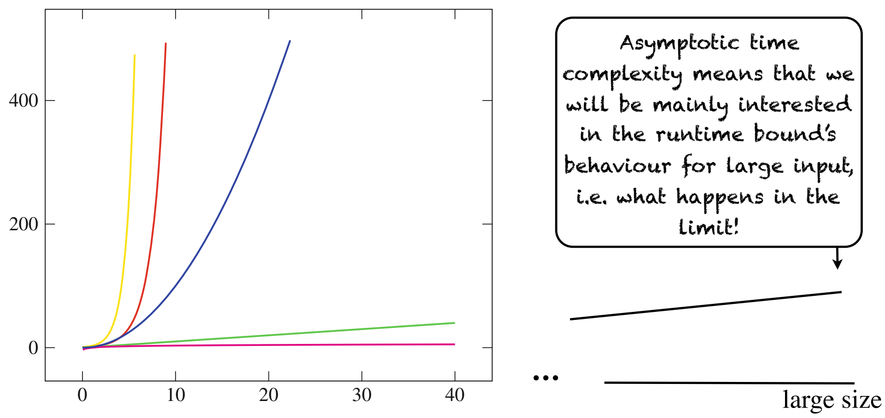
- A *runtime bound* is a total function on the natural numbers:

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

- which maps the size of the input of a program to the time it takes 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



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

size of input

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

functions with input *n* and output *p(n)*

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

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

Programs with Bounds (II)

3. $\mathtt{L}^{lintime} = \bigcup_{k \geq 0} \mathtt{L}^{time(k \times n)}$
   This set is the union of all sets of $\mathtt{L}$ programs that have a linear function f as time bound. *Linear functions* are of the form $f(n) = k \times n$.

4. $\mathtt{L}^{exptime} = \bigcup_{k \geq 0} \mathtt{L}^{time(2^{p(n)})}$
   Where $p(n)$ is a polynomial. This set is the union of all sets of $\mathtt{L}$ programs that have an exponential function f as time bound. Exponential functions are of the form $f(n) = 2^{p(n)}$ so we allow not only n in the exponent but any polynomial of n as well.

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*-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

Problem Classes

Definition 13.2 (*complexity classes*) Given a timed programming language L and a total function $f : \mathbb{N} \to \mathbb{N}$, we define four (complexity) classes of problems:

1. The class of *problems* L*-decidable* in time f is:
   $\mathbf{TIME}^{\mathrm{L}}(f) = \{A \subseteq \{0,1\}^* \mid A \text{ is decided by some } p \in \mathrm{L}^{time(f)}\}$
   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 $\mathbf{P}^{\mathrm{L}}$ of *problems* L*-decidable in polynomial time* is:
   $\mathbf{P}^{\mathrm{L}} = \{A \subseteq \{0,1\}^* \mid A \text{ is decided by some } p \in \mathrm{L}^{ptime}\}$
   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.

Problem Classes (II)

3. The class $\mathbf{LIN}^{\mathtt{L}}$ of *problems* $\mathtt{L}$*-decidable in linear time* is:
   $\mathbf{LIN}^{\mathtt{L}} = \{A \subseteq \{0,1\}^* \mid A \text{ is decided by some } p \in \mathtt{L}^{lintime}\}$
   In other words, this is the class of all problems (or sets) A about words over alphabet $\{0, 1\}$ that are decided by an $\mathtt{L}$-program with a runtime that is bounded by a linear function.

4. The class $\mathbf{EXP}^{\mathtt{L}}$ of *problems* $\mathtt{L}$*-decidable in exponential time* is:
   $\mathbf{EXP}^{\mathtt{L}} = \{A \subseteq \{0,1\}^* \mid A \text{ is decided by some } p \in \mathtt{L}^{exptime}\}$
   In other words, this is the class of all problems (or sets) A about words over alphabet $\{0, 1\}$ that are decided by an $\mathtt{L}$-program with a runtime that is bounded by an exponential function.

Lifting Simulation Properties to Classes

Lemma 13.1 $\mathtt{L} \preceq^{lintime} \mathtt{M}$ *implies* $\mathbf{LIN}^{\mathtt{L}} \subseteq \mathbf{LIN}^{\mathtt{M}}$*, and as a consequence,* $\mathtt{L} \equiv^{lintime} \mathtt{M}$ *implies* $\mathbf{LIN}^{\mathtt{L}} = \mathbf{LIN}^{\mathtt{M}}$.

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

Proof in exercises.

Similar results can be shown for **P**

END

Next time:
Robustness of $\mathbf{P}$