The Complexity Class P

valentinpi

4. Dezember 2020
(neueste Version)

Proseminar Theoretische Informatik WiSe 2020-21
Institut für Informatik
Freie Universität Berlin

Computability versus Complexity

Computability versus Complexity

What kind of efficiency?

Definition
Let $f : \mathbb{N} \to \mathbb{R}^+$ be a function. The complexity class *TIME* of f contains all languages that are computable in $\mathcal{O}(f(n))$ time for an input of size n.

$$\text{TIME}(f(n)) \coloneqq \{ L \mid \text{There is a deterministic TM that decides } L \text{ in } \mathcal{O}(f(n)) \text{ time for an input of size } n \}$$

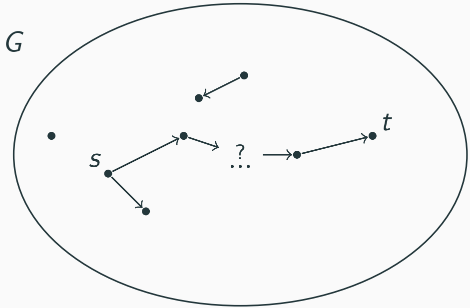
The Class P

Definition
P is the class of languages decidable in polynomial time on a deterministic single-tape TM.

$$\mathrm{P} := \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(n^k)$$

The PATH Problem

$$\text{PATH} \coloneqq \{\ \langle G, s, t\rangle\ | G \text{ is a directed graph and there is a path between nodes } s \text{ and } t\ \}$$



PATH $\in$ P

Theorem
PATH $\in$ P

Theorem
PATH $\in$ P

Beweis.

Input: $\langle G, s, t\rangle$, where G is a directed graph with nodes s, t.

Function:

1. Mark s.
2. Repeat until no more nodes are marked:
   2.1. Search through edges E. If an edge (u, v) is found with u marked and v not marked, mark v.
3. If t is marked, accept. Otherwise, reject.

$\blacksquare$

The RELPRIME Problem

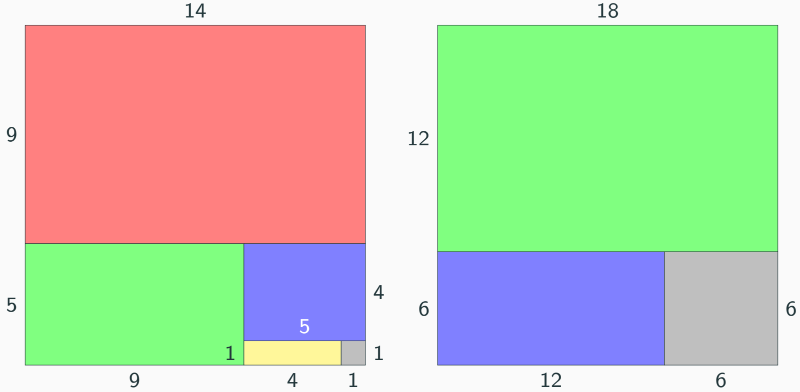
If two natural integers have the greatest common divisor (GCD) 1, they are called *relatively prime*.

The RELPRIME Problem

If two natural integers have the greatest common divisor (GCD) 1, they are called *relatively prime*.

$$\text{RELPRIME} := \{ \langle x, y \rangle \mid x \text{ and } y \text{ are relatively prime} \}$$

The Euclidian Algorithm



The RELPRIME Problem

Theorem
RELPRIME $\in$ P

The RELPRIME Problem

Theorem
RELPRIME $\in$ P

Beweis.

Input: $\langle x, y \rangle$, where $x, y \in \mathbb{N}$.

Function:

while $y > 0$ **do**
    $x \leftarrow x \bmod y$
    Swap x and y
If $x = 1$, accept. Otherwise, reject.

∎

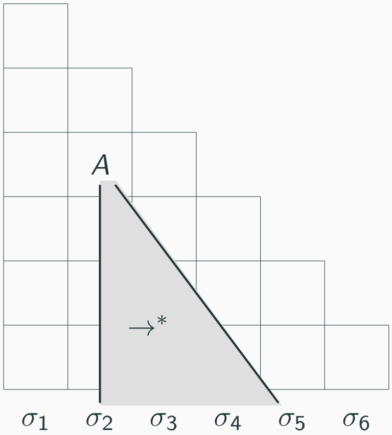
Context-Free Languages

Theorem
Every context-free language $L \in \mathrm{CFL}$ is in P.

$\Rightarrow$ CYK algorithm

Reminder: CYK Algorithm

Follows the principle of dynamic programming. Recursion from last semester: $V[i,j] = \{\ A \in V \mid A \rightarrow^* \sigma_i ... \sigma_j\ \}$



$CFL \subset$ **P**

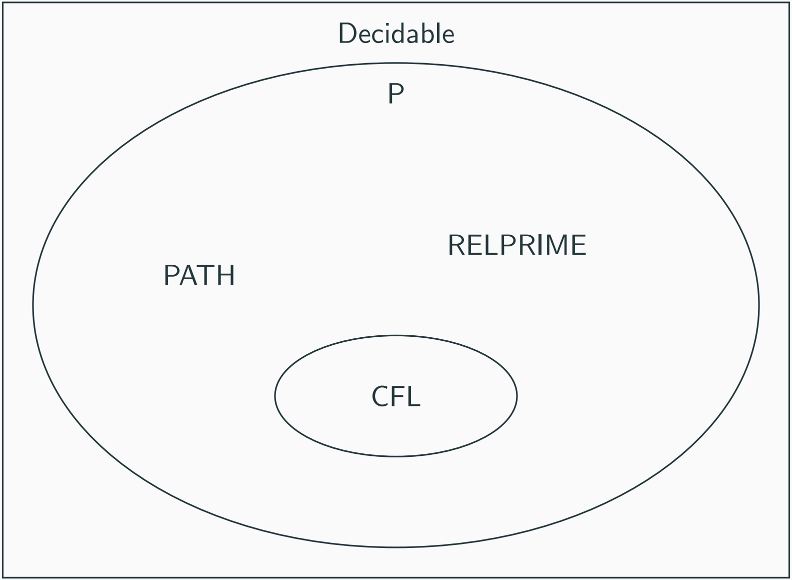
Beweis.

Input: $w \in \Sigma^*$, for $w \neq \varepsilon$ write $w = \sigma_1\sigma_2...\sigma_n$.

Function:

For $w = \varepsilon$, if $S \to \varepsilon$ is production, accept. Otherwise, reject.
for $i = 1$ to n **do**
    for each variable A **do**
        If $A \to \sigma_i$ is a rule, place A in *table*(i, i).
for $l = 2$ to n **do** ▷ Substring length
    for $i = 1$ to $n - l + 1$ **do** ▷ Starting position
        $j \leftarrow i + l - 1$ ▷ End position
        for $k = i$ to $j - 1$ **do** ▷ Split position
            for each rule $A \to BC$ **do**
                If $B \in$ *table*(i, k) and $C \in$ *table*$(k + 1, j)$, put A in *table*(i, j).
If $S \in$ *table*$(1, n)$, accept. Else, reject.

Final Landscape of Languages



Resources

Michael Sipser.
Introduction to the Theory of Computation, Third Edition.
Cengage Learning, 2012.

David Wees.
Visualizing euclid's algorithm.
`https://www.geogebra.org/m/ztbesvsd`, 11.11.2020, 21:30 (last visited).