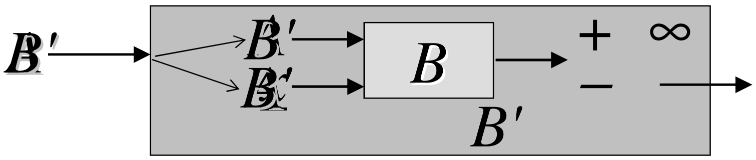
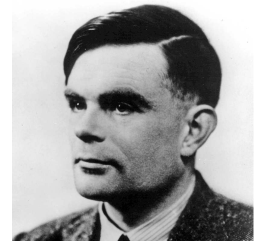
Alan M. Turing 1936

- first scientific calculations on digital computers
- *What are its fundamental limitations?*





- Undecidable <u>Halting Problem H</u>: **No** algorithm B can always correctly ans ~~simulator/interpreter B ?~~
Given $\langle A,\underline{x}\rangle$, does algorithm A terminate on input $\underline{x}$?

Proof by contradiction: Consider algorithm B' that, on input A, executes B on $\langle A,A\rangle$ and, upon a positive answer, loops infinitely. How does B' behave on B' ?

Un-/Semi-/Decidability I

Definition: a) An 'algorithm' $\mathcal{A}$ computes a partial function $f :\subseteq \mathbb{N} \to \mathbb{N}$ if it

- on inputs $\underline{x}\in\mathrm{dom}(f)$ prints $f(\underline{x})$ and terminates,
- on inputs $\underline{x}\notin\mathrm{dom}(f)$ does not terminate.

Injective <u>pairing function</u> ("*Hilbert Hotel*") $\langle x,y\rangle := x + (x+y)\cdot(x+y+1)/2$

b) $\mathcal{A}$ decides set $L\subseteq\mathbb{N}$ if it computes its total char. function: $\mathrm{cf}_L(\underline{x}):=1$ for $\underline{x}\in L$, $\mathrm{cf}_L(\underline{x}):=0$ for $\underline{x}\notin L$.

c) $\mathcal{A}$ semi-decides L if terminates precisely on $\underline{x}\in L$

d) $\mathcal{A}$ enumerates L if $L=\mathrm{range}(f)$ for some computable total injective $f:\mathbb{N}\to\mathbb{N}$.

Un-/Semi-/Decidability II

Example: The Halting problem H, <u>considered as subset of $\mathbb{N}$</u>, is semi-decidable, not decidable.

Theorem: a) Every finite L is decidable.
b) L is decidable iff its complement $\overline{L}$ is.
c) L is decidable iff both L, $\overline{L}$ are semi-decidable.
d) L is enumerable iff infinite and semi-decidable.

b) $\mathcal{A}$ decides set $L\subseteq\mathbb{N}$ if it computes its total char. function: $\mathrm{cf}_L(\underline{x}):=1$ for $\underline{x}\in L$, $\mathrm{cf}_L(\underline{x}):=0$ for $\underline{x}\notin L$.

c) $\mathcal{A}$ semi-decides L if terminates <u>precisely</u> on $\underline{x}\in L$

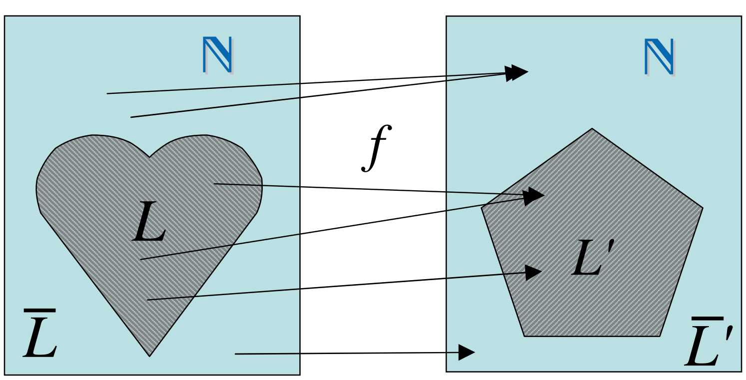
d) $\mathcal{A}$ enumerates L if $L=\mathrm{range}(f)$
for some computable total injective $f:\mathbb{N}\to\mathbb{N}$.

Comparing Decision Problems

Halting problem $H = \{ \langle\mathcal{A},\underline{x}\rangle : \mathcal{A}(\underline{x})$ terminates $\}$

Nontriviality $N = \{ \langle\mathcal{A}\rangle : \exists\underline{y}\ \mathcal{A}(\underline{y})$ terminates $\}$

Totality problem $T = \{ \langle\mathcal{A}\rangle : \forall\underline{z}\ \mathcal{A}(\underline{z})$ terminates $\}$



- $H \preccurlyeq N$ undecidable
- $H \preccurlyeq T$ undecidable
- $N \preccurlyeq H \not\preccurlyeq \overline{H}$
- $\overline{H} \preccurlyeq T \Rightarrow T \not\preccurlyeq H$

For $L,L'\subseteq\mathbb{N}$ write $L\preccurlyeq L'$ if there is a computable $f:\mathbb{N}\to\mathbb{N}$ such that $\forall\underline{x}:\ \underline{x}\in L \Leftrightarrow f(\underline{x})\in L'$.
a) $\overline{L'}$ semi-/decidable $\Rightarrow$ so $\overline{L}$. b) $L\preccurlyeq L'\preccurlyeq L'' \Rightarrow L\preccurlyeq L''$

WHILE+ Programs

$x_j := 0 \mid x_j := 1 \mid x_j := x_i + x_k \mid x_j := x_i \ominus x_k \mid$ shift
$x_j := x_i \oslash 2 \mid P;P \mid$ WHILE x_i DO P END

Syntax in Backus—Naur Form

Semantics: Input $x_1 \in \mathbb{N}$ or $(x_1,\ldots,x_d) \in \mathbb{N}^d$ or $\underline{x} \in \mathbb{N}^{\mathbb{N}}$
$x \ominus y = \max(0, x-y)$, $x \oslash 2 = \lfloor x/2 \rfloor$, $(x_1, x_2, \ldots) \rightarrow (x_2, x_3, \ldots)$
loop as long as $x_i \neq 0$, output$=x_0 \in \mathbb{N}$,

Definitions: binary *length* of $x \in \mathbb{N}$: $\ell(x) = \lceil \log_2(1+x) \rceil$

- time of a WHILE+ program P on input $\underline{x} = (x_1, \ldots x_d)$
- asymptotic time $t(n)$:
  worst-case over all inputs $\underline{x}$ with $\ell(\underline{x}) < n$

Asymptotic Runtime

$x_j := 0 \mid x_j := 1 \mid x_j := x_i + x_k \mid x_j := x_i \ominus x_k \mid$ shift
$x_j := x_i \oslash 2 \mid P;P \mid$ WHILE x_i DO P END

n	$\log_2 n \cdot 10$s	$n \cdot \log n$ sec	n^2 msec	n^3 µsec	2^n nsec
10	33sec	33sec	0.1sec	1msec	1msec
100	≈1min	11min	10sec	1sec	40 Mrd. Y
1000	≈1.5min	≈3h	17min	17min	
10 000	≈2min	1.5 days	≈1 day	11 days	
100 000	≈2.5min	19 days	4 months	32 years	

Definitions: binary *length* of $x \in \mathbb{N}$: $\ell(x) = \lceil \log_2(1+x) \rceil$

- time of a WHILE+ program P on input $\underline{x} = (x_1, \ldots x_d)$
- asymptotic time $t(n)$:
  worst-case over all inputs $\underline{x}$ with $\ell(\underline{x}) < n$

Some Complexity Classes

Definition: a) A WHILE+ program **computes** the function $f:\mathbb{N}\to\mathbb{N}$ if on input x it prints $f(x)$ and terminates in time $t(n)$ $n:=\ell(\underline{x})$

Polynom.growth: $\exists k\ t(n)\leq O(n^k)$; exponential: $2^{O(n^k)}$

Def: For decision problems $L\subseteq\mathbb{N}$

- $\mathcal{P}$ = { L decidable in polynomial time }
- $\mathcal{NP}$ = { L verifiable in polynomial time }, i.e.

$L=\{\ x\in\mathbb{N} : \exists y\in\mathbb{N},\ \ell(y)\leq\mathrm{poly}(\ell(x)),\ \langle x,y\rangle\in V\ \},\ \ V\in\mathcal{P}$

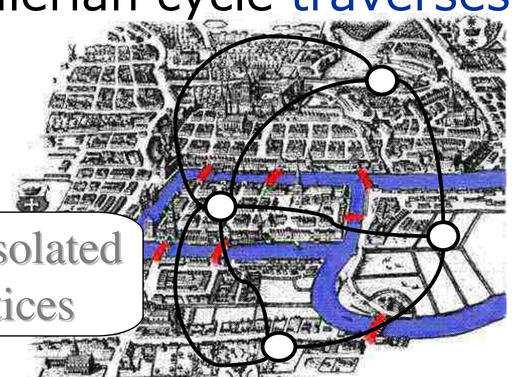
- $\mathcal{EXP}$ = { L decidable in exponential time }

Theorem: $\mathcal{P}\subseteq\mathcal{NP}\subseteq\mathcal{EXP}$

Example Decision Problems

In an undirected graph *G*, Eulerian cycle traverses each <u>edge</u> precisely once;

Hamiltonian cycle visits each <u>vertex</u> precisely once.



G admitting a Eulerian cycle is connected and has an even number of edges incident to each vertex

save isolated vertices

Theorem: Conversely every connected graph with an even number of edges incident to each vertex admits a Eulerian cycle.

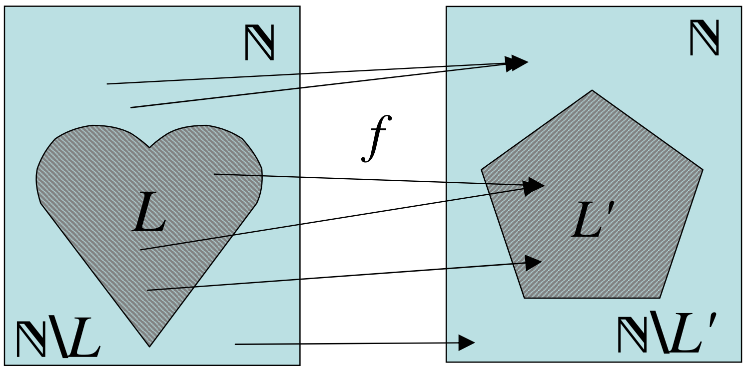
EC := { $\langle G\rangle$ | G has a Eulerian cycle} $\mathcal{NP}$

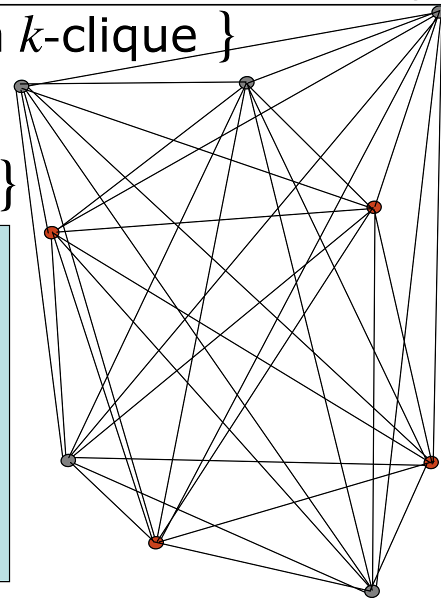
HC := { $\langle G\rangle$ | G has Hamiltonian cycle} $\mathcal{NP}$

Comparing Decision Problems 2

CLIQUE = { $\langle G,k\rangle$ | G contains a k-clique }

$\equiv_p$ **IS**={ $\langle G,k\rangle$: G has k pairwise non-connected vertices}

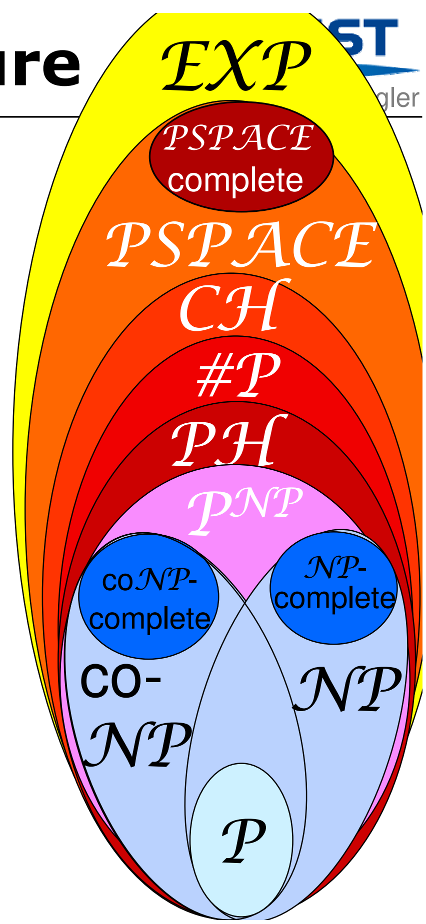




For $L,L'\subseteq\mathbb{N}$ write $L\preccurlyeq_p L'$ if exists a polynomial-time computable f: $\mathbb{N}\to\mathbb{N}$ such that $\forall\underline{x}$: $\underline{x}\in L \Leftrightarrow f(\underline{x})\in L'$

Lemma: a) $L\preccurlyeq_p L'\preccurlyeq_p L'' \Rightarrow L\preccurlyeq_p L''$ b) $L'\in\mathcal{P} \Rightarrow L\in\mathcal{P}$

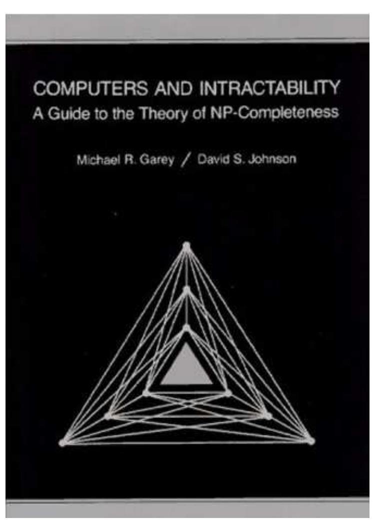
Complexity Class Picture



Def: $A\in\mathcal{NP}$ is $\mathcal{NP}$-complete if $L\preccurlyeq_p A$ holds for every $L\in\mathcal{NP}$.

Theorem (Cook'72/Levin'71): **SAT** is $\mathcal{NP}$-complete!

Lemma: For A $\mathcal{NP}$-complete and $A\preccurlyeq_p B\in\mathcal{NP}$, B is also $\mathcal{NP}c$.



Now know ≈500 natural problems $\mathcal{NP}$-complete…