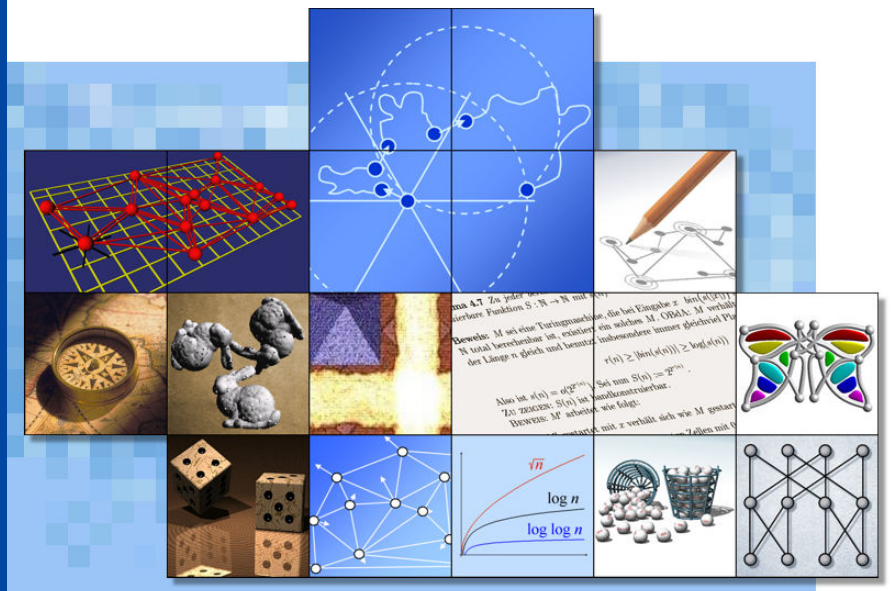
HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

CS493 – Special Topics in Computer Science:

Diagonalization



Martin Ziegler

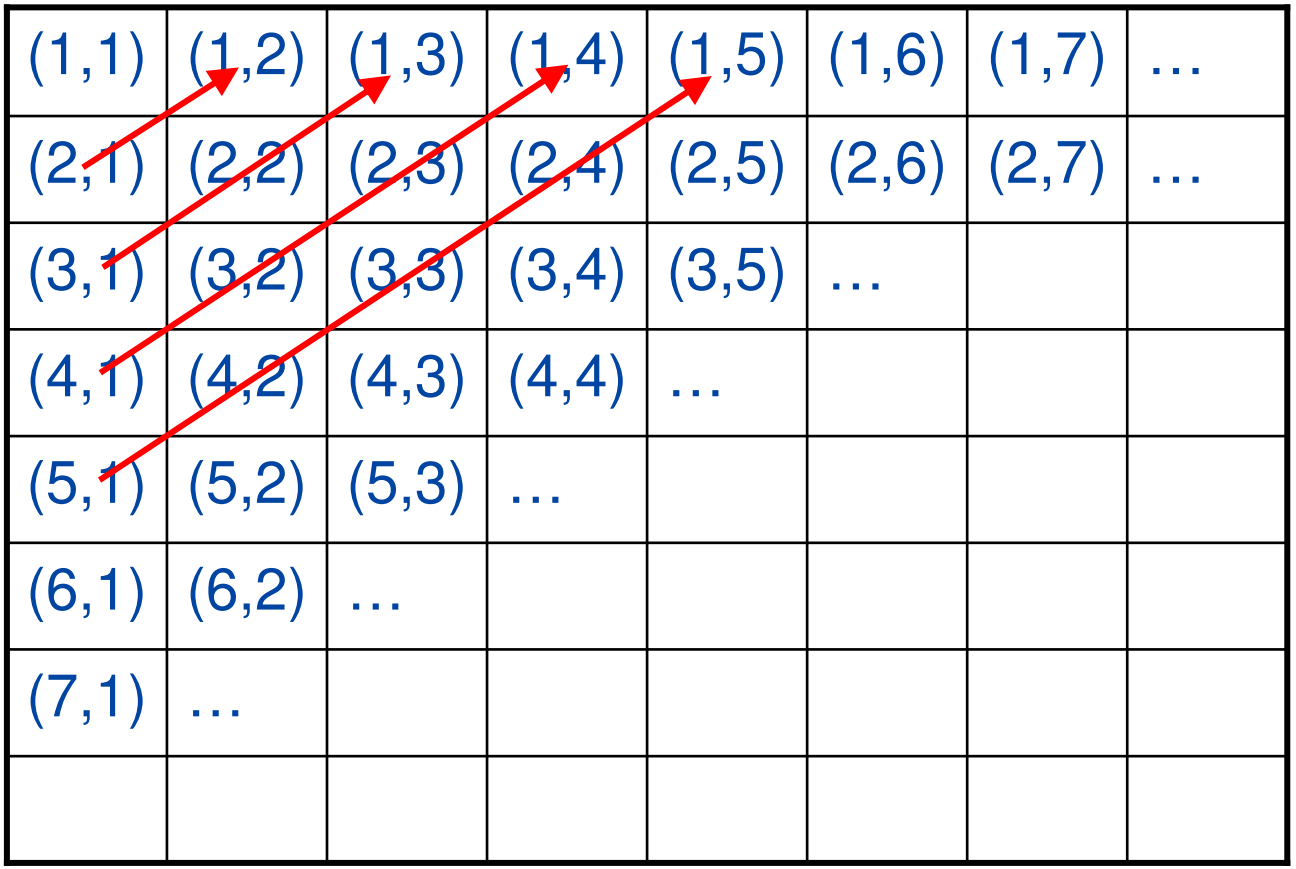
and 정 지원

KAIST

Cantor

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Card(ℕ) = Card(ℕ×ℕ):



Cantor

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

$\mathrm{Card}(\mathbb{N}) = \mathrm{Card}(\mathbb{N}\times\mathbb{N}) = \mathrm{Card}(\mathbb{N}^*$, all finite sequences)

$= \mathrm{Card}(2^{\mathbb{N}}$, all infinite binary sequences) ?

Suppose yes:

$\exists$binary sequence $(b_{n,m})$ enumerating all bin. sequences,

i.e. $\forall(c_m)$

$\exists n$: $b_{n,m} = c_m$.

New sequence

$c_m := (1\text{-}b_{m,m})$

#1	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	…
#2	b_{21}	b_{22}	b_{23}	b_{24}	b_{25}	b_{26}	…
#3	b_{31}	b_{32}	b_{33}	b_{34}	b_{35}	b_{36}	…
#4	b_{41}	b_{42}	b_{43}	b_{44}	b_{45}	b_{46}	…
#5	b_{51}	b_{52}	b_{53}	b_{54}	b_{55}	b_{56}	…
#6	b_{61}	b_{62}	b_{63}	b_{64}	b_{65}	b_{66}	…
…	…	…	…	…	…	…	…

Cantor

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

$\mathrm{Card}(\mathbb{N}) = \mathrm{Card}(\mathbb{N}\times\mathbb{N}) = \mathrm{Card}(\mathbb{N}^*$, all finite sequences)

$\neq \mathrm{Card}(2^{\mathbb{N}}$, all infinite binary sequences) ?

Suppose yes:

$\exists$binary sequence $(b_{n,m})$ enumerating all bin. sequences,

i.e. $\forall(c_m)$

$\exists n$: $b_{n,m} = c_m$.

New sequence

$c_m := (1\text{-}b_{m,m})$

#1	$-b_{11}$	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	…
#2	b_{21}	$-b_{22}$	b_{23}	b_{24}	b_{25}	b_{26}	…
#3	b_{31}	b_{32}	$-b_{33}$	b_{34}	b_{35}	b_{36}	…
#4	b_{41}	b_{42}	b_{43}	$-b_{44}$	b_{45}	b_{46}	…
#5	b_{51}	b_{52}	b_{53}	b_{54}	$-b_{55}$	b_{56}	…
#6	b_{61}	b_{62}	b_{63}	b_{64}	b_{65}	$-b_{66}$	…
…	…	…	…	…	…	…	…

Limits of Computability

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Def: A decision problem is a set $L \subseteq \{0,1\}^*$

An algorithm decides L if,

- on inputs $x \in L$, it terminates and outputs 1;
- on inputs $x \notin L$, it terminates and outputs 0.

(Disregard optimization+search problems...)

What is an algorithm? Next slide. Now suffices

Each algorithm encodable as finite sequence

→ There are countably many algorithms

but uncountably many decision problems:

Most *L* are algorithmically unsolvable!

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

What is an algorithm?

```
Algorithm unique(n,x)
// Are x[1],…,x[n] pairwise distinct?
qsort(n,x);
for i:=1 to n-1 do
 if x[i]=x[i+1] return false;
endfor;
return true
```

```
Algorithm get_a_life;
Meet nice girl/boy X;
Find nice restaurant R;
Invite X to date in R;
Change gravitational constant
                of the universe.
See what happens.
```

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

What is an algorithm?

And how much resources does it consume?

- running time
- memory
- #processors
- communication
- …

Algorithm IndependentSet(V,E,k)
// Does graph (V,E) have k pair-
// wise non-connected vertices ?
Declare Boolean variables $x[v,i]$,
$v \in V,\ i=1,\ldots,k$
Define Boolean formulas

$\varphi_i := \bigvee_{v \in V} x[v,i], \qquad i=1\ldots k$

$\psi_{v,i,j} := \neg x[v,i] \vee \neg x[v,j],$
$v \in V,\ 1 \le i < j \le k$

$\chi_{u,v,i,j} := \neg x[u,i] \vee \neg x[v,j],$
$\{u,v\} \in E,\ 1 \le i < j \le k$

return SatisfiableKNF
$(\{\varphi_i, \psi_{v,i,j}, \chi_{u,v,i,j} : u,v,i,j\}, x)$

Notions of Computation

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

Def: A partial function $f:\subseteq \mathbb{N}^* \to \mathbb{N}$ is computable

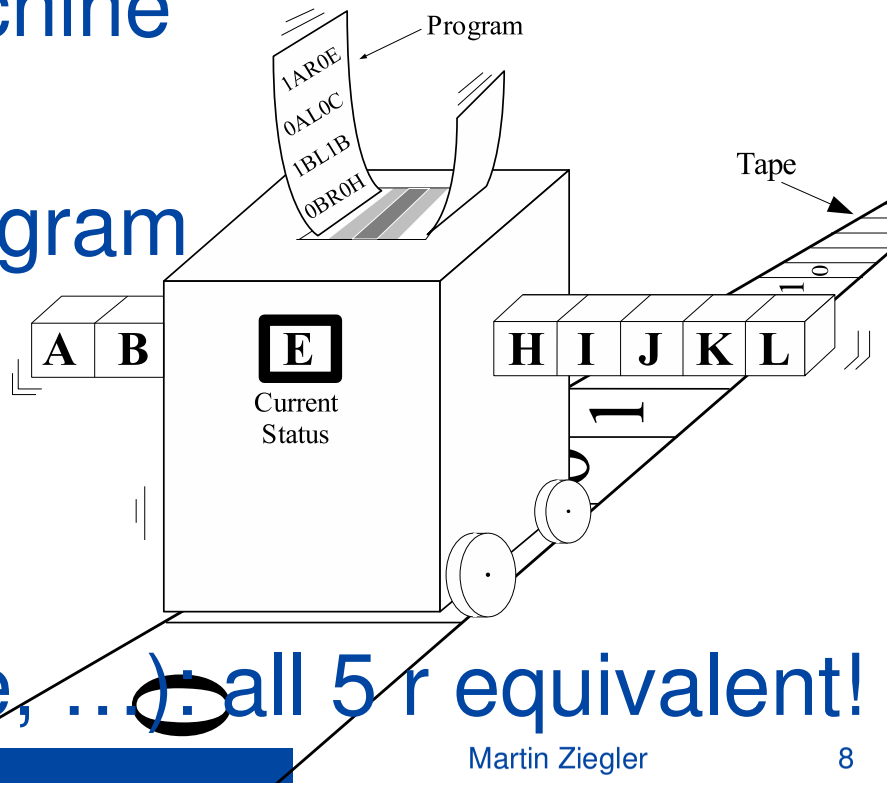
a) iff there exists a **`WHILE`**-program computing *f*

b) iff there exists a λ-program computing *f*

c) iff there is a Turing machine computing *f* (in binary)

d) iff there exists a **`BF`**-program computing *f* (in binary)

e) iff *f* is μ-recursive.



Theorem (Church, Kleene, …): all 5 r equivalent!

The Language BF

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

- By the above theorem, it does not matter which of the 5 programming systems to work in.
- We shall occasionally refer to **BF** in this lecture
- in order to exemplify some abstract concepts,
- that is, for purely didactical reasons:
- Nothing really depends logically on it!
- E.g. UTM property: there is a universal program U which, given (an encoding of) any other program P and input x, simulates P on x.
- Or SMN property: Given program P and input x, can algorithmically construct program $Q=Q(P,x)$ which, on any input y, behaves like P on x.

Semi-/Decidability and Enumerability

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

A set $L \subseteq \{0,1\}^*$ is decided by program P if

- on inputs $x \in L$, P terminates and outputs 1;
- on inputs $x \notin L$, P terminates and outputs 0.

L is semi-decided by P if

- on inputs $x \in L$, P terminates;
- on inputs $x \notin L$, P does not terminate.

P enumerates L if

- on every input x, P outputs some $f(x) \in L$;
- each $y \in L$ is the output $y = f(x)$ of some x.

Semi-/Decidability and Enumerability

HEINZ NIXDORF INSTITUTE
University of Paderborn
Algorithms and Complexity

a) If L is decidable, then so is its complement
b) If L is decidable, then it is semi-decidable.
c) If $L \neq \emptyset$ is semi-decidable, L is enumerable:
   - Fix $z \in L$. Given $x = \langle y,n \rangle$, simulate (UTM) semi-deciding "$y \in L$?" for n steps. If terminates, output y, else output z.
d) If L is enumerable, L is semi-decidable:
   - Given z, test "$z = f(x)$" for each x.
e) If both L and its complement are semi-decidable, then L is decidable:
   - Given x, semi-decide "$x \in L$" and L's complement simultaneously (dovetailing)