# §2 Basic Computability Theory



- Computability, semi-/decidability, enumerability
- Examples of undecidable problems
- Reduction: comparing problems
- (Busy Beaver function)
- LOOP programs
- Ackermann function

# 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 <u>precisely</u> on $\underline{x} \in L$

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

# Un-/Semi-/Decidability II

**Example:** The Halting problem $H$, considered as subset of $\mathbb{N}$, 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 precisely on $\underline{x} \in L$

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

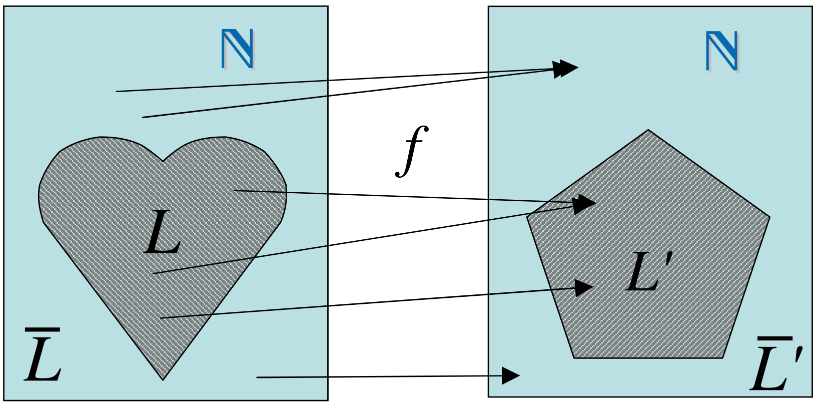


# Comparing Decision Problems

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

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

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




- $H \preccurlyeq N$ 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) $L \preccurlyeq L' \preccurlyeq L'' \Rightarrow L \preccurlyeq L''$

# LOOP Programs



**Syntax** in Backus—Naur Form:

$P := (\ x_j := 0 \mid x_j := x_i + 1 \ \mid\ P\,;P \mid$
$\quad \text{LOOP}\ x_i\ \text{DO}\ P\ \text{END}\ )$ $\quad i,j \in \mathbb{N}$

**Semantics:**

- $x_1,\ldots x_k$ contain input $\in\mathbb{N}^k$
- LOOP executed $x_i$ times
- Body must not change $x_i$

**Example:** simulate "if $x_j \neq 0$ then $P$ else $Q$"

$x_k := 0\ ;\ \text{LOOP}\ x_j\ \text{DO}\ x_k := 1\ \text{END}\ ;\ x_\ell := 1;$
$\text{LOOP}\ x_k\ \text{DO}\ P\ ;\ x_\ell := 0\ \text{END}\ ;\ \text{LOOP}\ x_\ell\ \text{DO}\ Q\ \text{END}$

**Example:** simulate "$x_j := \max(0, x_i - 1)$" :

$x_j := 0\ ;\ x_k := 0\ ;$
$\text{LOOP}\ x_i\ \text{DO}$
$\quad x_j := x_k\ ;\ x_k := x_k + 1$
$\text{END}$

# Capabilities of LOOP Programs

KAIST CS422 M. Ziegler

**Examples:** simulate addition "$x_k := x_j + x_i$"

$x_k := 0;\ \text{LOOP}\ x_j\ \text{DO}\ x_k := x_k + 1\ \text{END};$
$\text{LOOP}\ x_i\ \text{DO}\ x_k := x_k + 1\ \text{END}$

Simulate multiplication "$x_k := x_j \times x_i$"

$x_k := 0;\ \text{LOOP}\ x_i\ \text{DO}\ x_k := x_k + x_j\ \text{END}$

Now recall Ackerman's function $\quad A(0,n) = 2+n$

$A(\ell+1,0) = 1 \qquad A(\ell+1,n+1) = A(\ell, A(\ell+1,n))$

**Theorem:** • To every LOOP program $P = P(x_1,\ldots x_k)$ there exists some $\ell = \ell(P) \in \mathbb{N}$ s.t. $P$ on input $\underline{x} \in \mathbb{N}^k$ makes $\leq A(\ell,n) < \infty$ steps, where $n := \max(2, \sum_j |x_j|)$

• $A(n,n)$ is <u>not</u> computable by <u>any</u> LOOP program!

# Proof by Structural Induction

KAIST
CS422 M. Ziegler

$P := (\ x_j := 0 \mid x_j := x_i + 1 \ \mid\ P\,;P \mid\ \text{LOOP } x_i \text{ DO } P \text{ END}\ )$

**Lemma:** $A(\ell{+}1,n{+}m) = A(\ell,A(\ell,A(\ldots A(\ell,A(\ell{+}1,n)))))$

**Proof**, induction: $x_j := 0 \mid x_j := x_i + 1$: $1 \leq A(1,1)$ steps

$P\,;P$ : $A(\ell,n) + A(\ell,A(\ell,n)) \leq A(\ell,n) + A(\ell{+}1,n{+}1)$ $\leq A(\ell{+}2,n)$ steps

LOOP $x_i$ DO $P$ END:

$A(\ell,n\text{-}x_i)+A(\ell,A(\ell,n\text{-}x_i))+A(\ell,A(\ell,A(\ell,n\text{-}x_i)))+\ldots$ [$x_j$-iter.] steps

$\leq A(\ell{+}1,n)+A(\ell{+}1,n)+\ldots$ $\leq A(\ell{+}2,n)$

**Theorem:** • To every LOOP program $P=P(x_1,\ldots x_k)$ there exists some $\ell=\ell(P)\in\mathbb{N}$ s.t. $P$ on input $\underline{x}\in\mathbb{N}^k$ makes $\leq A(\ell,n)<\infty$ steps, where $n:=\max(1,\|\underline{x}\|_1)$

$A(\ell{+}1,n)=2+n$, $A(\ell{+}1,0)=1$, $A(\ell{+}1,n{+}1) = A(\ell,A(\ell{+}1,n))$

# Power of LOOP Programs 2

KAIST
CS422 M. Ziegler

**Def:** Recall bijective $\mathbb{N}^2\ni(x,y)\to\langle x,y\rangle:=2^x\cdot(2y+1)\text{-}1\in\mathbb{N}$ and write $\langle x,y,z\rangle:=\langle\langle x,y\rangle,z\rangle$, $\langle x,y,z,w\rangle:=\langle\langle x,y,z\rangle,w\rangle$ etc.

**Lemma: a)** There exists a LOOP program that, given $x,y\in\mathbb{N}$, returns $\langle x,y\rangle\in\mathbb{N}$.

**b)** There exists a LOOP program that, given $\langle x,y\rangle\in\mathbb{N}$, returns $x$ and $y\in\mathbb{N}$.

**c)** There exists a LOOP program that, given integers $n\leq N$ and $\langle x_1,x_2,\ldots,x_n,\ldots,x_N\rangle$, returns $x_n$.

**d)** There exists a LOOP program that, given $n\leq N$ and $y$ and $\langle x_1,x_2,\ldots,x_n\ldots,x_N\rangle$, returns $\langle x_1,x_2,\ldots,y,\ldots,x_N\rangle$.

array of integers with indirect addressing