

# Turing Machines as a Computer

- A function  $f : \Sigma^* \rightarrow \Sigma^*$  is *computable* if some TM  $M$  exists which on every input  $w$  halts with  $f(w)$  on the tape.
- TM for a computable function halts on every input, like Turing deciders.
- **Example:** All usual arithmetic functions on integers are **computable**:  $\langle m, n \rangle \mapsto m + n$ ,  $\langle m, n \rangle \mapsto m * n$ ,  $\langle m, n \rangle \mapsto m^n$ .
- Using recursion and subroutines.

# 22c:131 Limits of Computation

Hantao Zhang

<http://www.cs.uiowa.edu/~hzhang/c131>

The University of Iowa  
Department of Computer Science

Mapping and Turing Reducibilities

# Mapping Reducibility

- To reduce a problem  $A$  to a problem  $B$  by mapping reducibility means to find a computable function,  $f : A \rightarrow B$  called a *reduction*, that converts instances of  $A$  into instances of  $B$ , such that for any  $w \in \Sigma^*$ ,  $w \in A$  iff  $f(w) \in B$ .
- Computable functions may be transformations of machine descriptions. A computable function  $f$  may take an input  $w$ , where  $w$  is an encoding of a TM  $M$ , i.e.,  $w = \langle M \rangle$ , and may return the description of another TM,  $\langle M' \rangle = f(w)$

# More Examples

- $\lfloor m/n \rfloor$
- $\lfloor \sqrt{(n)} \rfloor$
- $\lfloor \log_m(n) \rfloor$

## Application

Proof of the undecidability of PCP contains two mapping reductions:

- $A_{TM} \leq_m MPCP$  by  $f_1 : A_{TM} \rightarrow MPCP$
- $MPCP \leq_m PCP$  by  $f_2 : MPCP \rightarrow PCP$
- $A_{TM} \leq PCP$  by  $f = f_2(f_1(\langle M, w \rangle))$

## Some Known Results

- (Theorem 5.22) If  $A \leq_m B$  and  $B$  is decidable, then  $A$  is decidable.
- (Corollary 5.23) If  $A \leq_m B$  and  $A$  is undecidable then  $B$  is undecidable.
- (Theorem 5.28) If  $A \leq_m B$  and  $B$  is Turing-recognizable, then  $A$  is Turing-recognizable.
- (Corollary 5.29) If  $A \leq_m B$  and  $A$  is not Turing-recognizable, then  $B$  is not Turing-recognizable.

## Turing Reducibility

**Definition 6.18** An *oracle* for a language  $B$  is an external device that is capable of reporting whether a string  $w$  is a member of  $B$

**Oracle TM:** is a modified TM that has the additional capability of querying an oracle.

**Notation:**  $M^B$  is an oracle TM for language  $B$ .

## Intuitive Meaning of Reducibility

Consider two languages  $A_{TM}$  and  $\overline{A_{TM}}$ .

- Intuitively, they are reducible to each other because a solution to either could be used to solve the other by simply reversing the answer.
- However, we know that  $\overline{A_{TM}}$  is not mapping reducible to  $A_{TM}$  because  $A_{TM}$  is Turing recognizable but  $\overline{A_{TM}}$  is not.

## Procedure $T^{A_{TM}}$

$T^{A_{TM}}$  = "On input  $\langle M \rangle$  where  $M$  is a TM:

1. Construct the following TM  $N$ :  
 $N$  = "On any input:
  - (a) Run  $M$  in parallel on all strings in  $\Sigma^*$
  - (b) If  $M$  accepts any of these strings, *accept*"
2. Query the oracle to determine whether  $\langle N, 0 \rangle \in A_{TM}$ .
3. If the oracle answers *NO*, *accept*; if answers is *YES*, *reject*."

## Example

Consider an oracle for  $A_{TM}$

- An oracle TM with an oracle for  $A_{TM}$  can decide more languages than an ordinary TM can.
- Such a TM can obviously decide  $A_{TM}$  itself by querying the oracle about its input.
- Such a TM can also decide  $E_{TM}$  with the following procedure  $T^{A_{TM}}$ .

## Theorem 6.21

If  $A \leq_T B$  and  $B$  is decidable then  $A$  is decidable.

**Proof:** If  $B$  is decidable then we may replace the oracle for  $B$  by an actual procedure that decide  $B$ . Thus, we may replace the oracle TM that decides  $A$  by an ordinary TM that decides  $A$ .

## Definition 6.20

We say that  $E_{TM}$  is *decidable relative to*  $A_{TM}$ .

Language  $A$  is *Turing reducible* to language  $B$ ,  $A \leq_T B$ , is  $A$  is decidable relative to  $B$ .

Example: We have shown that  $E_{TM} \leq_T A_{TM}$

**Note:** Turing reducibility satisfies our intuitive concept of reducibility.

## Problem 6.4

Let  $A'_{TM} = \{\langle M, w \rangle \mid M \text{ is an oracle TM and } M^{A_{TM}} \text{ accepts } w\}$ .

Show that  $A'_{TM}$  is undecidable relative to  $A_{TM}$ .

## Note

- Turing reducibility is a generalization of mapping reducibility.
- If  $A \leq_m B$  then  $A \leq_T B$  because the mapping reduction may be used to give an oracle TM that decide  $A$  relative to  $B$ .
- An oracle TM with an oracle for  $A_{TM}$  can solve problems that are not solvable by ordinary TM.
- However, oracle Turing machines cannot decide all languages (Problem 6.4).