1 Mapping reducibility

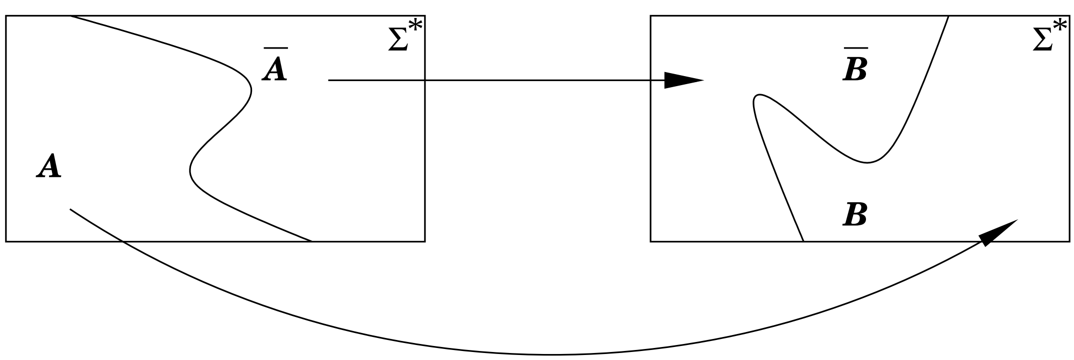
Definition 1 *A function* $f : \Sigma^* \rightarrow \Sigma^*$ *is a* **computable function** *if there is a TM which on every input* w *halts with just* $f(w)$ *on the tape.*

Example 1

- Usual arithmetic functions, *i.e.* addition, multiplication, etc are computable.
- Functions that **transform** descriptions of TMs:

Definition 2 *Language* A *is* **mapping reducible** *to language* B*, written* $A \leq_m B$*, if there is a computable function* $f : \Sigma^* \rightarrow \Sigma^*$*, where for every* w

$$w \in A \text{ if and only if } f(w) \in B.$$



Proposition 1 *If* $A \leq_m B$*, then* $\overline{A} \leq_m \overline{B}$.

Proposition 2 *If* $A \leq_m B$ *and* B *is decidable, then* A *is decidable. If* $A \leq_m B$ *and* A *is undecidable, then* B *is undecidable.*

Proposition 3 *If* $A \leq_m B$ *and* B *is recognizable, then* A *is recognizable. If* $A \leq_m B$ *and* A *is unrecognizable, then* B *is unrecognizable.*

Observation: Usually, if a problem A can be reduced to a problem B, there is a mapping reducibility from A to B.

Example 2

There is a mapping reduction f from A_{TM} to $HALT_{TM}$. The following TM F computes f:

On input $\langle M, w \rangle$;
construct a new TM M' by
on input x
run M on x
if M accepts
accept
if M rejects
enter an infinite loop
$f(\langle M, w \rangle) = \langle M', w \rangle$
/* M accepts w iff M' halts on w */

Conclusion: $HALT_{TM}$ is undecidable since A_{TM} is undecidable.

Example 3

There is a mapping reduction $f : E_{TM} \rightarrow EQ_{TM}$.

On input $\langle M \rangle$;
construct a new TM M' which rejects all inputs;
The mapping reduction f is defined by

$$f(M) = \langle M, M' \rangle.$$

/* Notice the property: $L(M) = \emptyset$ iff $L(M) = L(M')$ */

Conclusion: EQ_{TM} is undecidable since E_{TM} is undecidable.

Example 4

There is a mapping reduction $f : A_{TM} \rightarrow \overline{E}_{TM}$.

On input $\langle M, w \rangle$;
construct a new TM M' by
on input x
if $x \neq w$
REJECT
else
run M on w
ACCEPT if M accepts w
$f(\langle M, w \rangle) = \langle M' \rangle$

/* Thus M accepts w iff M' doesn't accept any string.*/

Conclusion: Since A_{TM} is undecidable, $\overline{E}_{TM}$ is also undecidable. Therefore E_{TM} is undecidable.

Theorem 1

EQ_{TM} is neither Turing-recognizable nor co-Turing-recognizable.

Proof. We construct two mapping reductions:

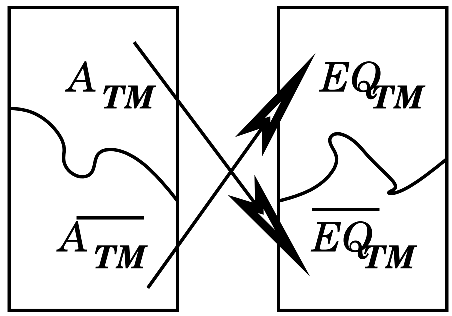
$$f : A_{TM} \to \overline{EQ}_{TM} \text{ and } g : A_{TM} \to EQ_{TM}$$

	mapping reduction f		mapping reduction g
On	input $\langle M, w\rangle$;	On	input $\langle M, w\rangle$;
	construct new TM M_1, M_2 by		construct new TM M_1, M_2 by
	M_1 : on any input		M_1 : on any input
	REJECT		ACCEPT
	M_2 : on any input		M_2 : on any input
	run M on w		run M on w
	ACCEPT if M accepts		ACCEPT if M accepts

Since f is a mapping reduction $A_{TM} \to \overline{EQ}_{TM}$, it is also a mapping reduction $\overline{A}_{TM} \to EQ_{TM}$. Hence, if EQ_{TM} were Turing-recognizable, the existence of f would prove that $\overline{A}_{TM}$ would be Turing-recognizable, implying that A_{TM} is decidable, which was proved to be wrong.

Similarly, the existence of a mapping reduction g implies that if $\overline{EQ}_{TM}$ were Turing-recognizable, then $\overline{A}_{TM}$ would be Turing-recognizable as well, implying that A_{TM} is decidable, which was proved to be wrong.

reduction f



reduction g