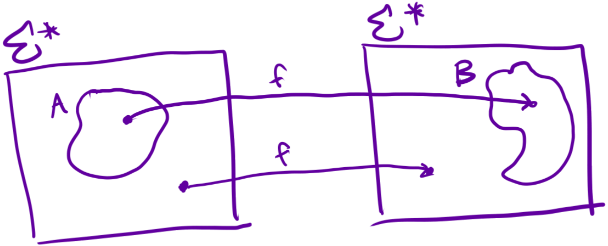
Mapping Reductions

<u>Definition:</u>

A is mapping reducible to B (written as $A \leq_m B$) means that there is a computable function $f: \Sigma^* \to \Sigma^*$ such that for all strings x in Σ^*,

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

If $x \in A$, then $f(x) \in B$
If $x \notin A$, then $f(x) \notin B$



Convert the question about x's membership in A to a question about f(x)'s membership in B:

- if $f(x) \in B$, then $x \in A$
- if $f(x) \notin B$, then $x \notin A$

If $A \leq_m B$, then A is no harder than B.

A is easier than B or A is equally as hard as B

A function $f: \Sigma^* \to \Sigma^*$ is a <u>computable function</u> if there exists some TM M such that each input x on M halts with exactly $f(x)$, followed by all blanks, on the tape.

Example:

The function that maps a string to the result of repeating the string twice.

Implementation level

$$f_2 : \Sigma^* \to \Sigma^* \qquad f_2(x) = xx$$

Define a TM:

Read x, write x'. Move right until $\sqcup$ is read, write #
Move left until x' is read. write x.

Read x. Write x'. Move right to the first blank and write x
Move left until x' is read, write x.
Move 1 to the right
repeat until x is #

Move every symbol to the right of # 1 spot to the left
Stop after a blank symbol is copied

$\Sigma = \{0, 1\}$ $\quad \Sigma' = \{a' \mid a \in \Sigma\}$

0	1	0	0	1	0					...

$\Sigma' = \{0', 1'\}$

$A \le_m B$ means that A is no harder than B

$\Sigma = \{0, 1\}$

$A = L(\Sigma^*)$

$B = \{ww \mid w \text{ is a string over } \Sigma\}$

$A \le_m B$??? ← yes! f_2 is a computable function where $x \in A$ iff $f_2(x) \in B$