4 Computability

Encoding of a multi-tape Turing Machine: *Assume our encoding of the TMs satisfy the following properties*

1. *Every string* $\alpha \in \{0,1\}^*$ *represents some TM (On invalid encoding* α, M_α *always reject)*
2. *Every Turing Machine is represented by infinitely many strings*

   TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

***thm* 4.1.** *(Universal Turing Machine)*

There exists a multitape Turing Machine $\mathcal{U}$, *s.t.* $\forall x, \alpha \in \{0,1\}^*, \mathcal{U}(x,\alpha) = M_\alpha(x)$. *Moreover, if* M_α *halts on input* x *within* T *steps, then* $\mathcal{U}(x,\alpha)$ *halts in* $O_M(T \log T)$ *steps (weaker version* $O_{M_\alpha}(T(n))^2$*)*

***lemma* 4.2.** *Almost all languages are undecidable*

Proof. $\#languages = 2^{\aleph_0} = \aleph_1$

$\#\{L \subseteq \{0,1\}^*\}$

$\#TMs = \aleph_0$ □

thoughts: ***diagonalization***

$def\ L_{flip} = \{\alpha : M_\alpha \text{ does not accept } \alpha\}$

***lemma* 4.3.** L_{flip} *is undecidable*

Proof. *Assume for contradiction that* L_{flip} *is decided by a TM* M_β, *which implies that* $L(M_\beta) = L_{flip}$

- *case1:* $\beta \in L_{flip}$. *By definition* M_β *does not accept* β,*i.e,* M_β *rejects* β. *So,* $\beta \notin L(M_\beta) = L_{flip}$. *Contradiction!*
- *case2:* $\beta \notin L_{flip}$. *By definition,* M_β *accepts* β. *So,* $\beta \in L(M_\beta) = L_{flip}$. *Contradiction!*

□

Turing halting problem

$L_{halt} = \{(\alpha, x), M_\alpha \text{ halts on } x\}$

Fermat's Last Theorem

$(\forall m \geq 3)(\forall a, b, c \geq 1)(a^m + b^m \neq c^m)$

$$M_\alpha \begin{cases} T = 2 \\ while\ true \\ T = T + 1 \\ for\ d = 3\ to\ T \\ for\ a, b, c \in \{1, 2, ..., T\} \\ if(a^d + b^d = c^d) \to exit \end{cases}$$

FLT iff $(M_\alpha, \varepsilon) \notin L_{halt}$

****reduction***

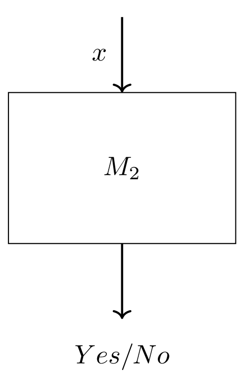
***def* 4.4.** *Let* $L_1, L_2 \subseteq \{0,1\}^*$. *Write* $L_1 \leq L_2$ *if there is a reduction from* L_1 *to* L_2 *that, there exists a TM* $M : \{0,1\}^* \to \{0,1\}^*$ *(On any input x, M always halts and outputs a string M(x)), s.t.*

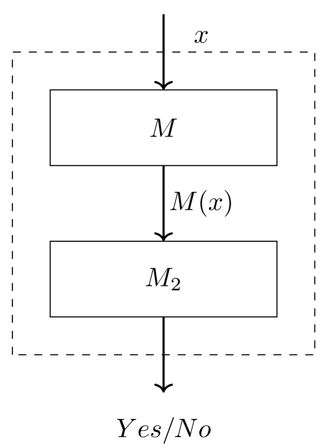
1. $(\forall x \in L_1)(M(x) \in L_2)$
2. $(\forall x \notin L_1)(M(x) \notin L_2)$

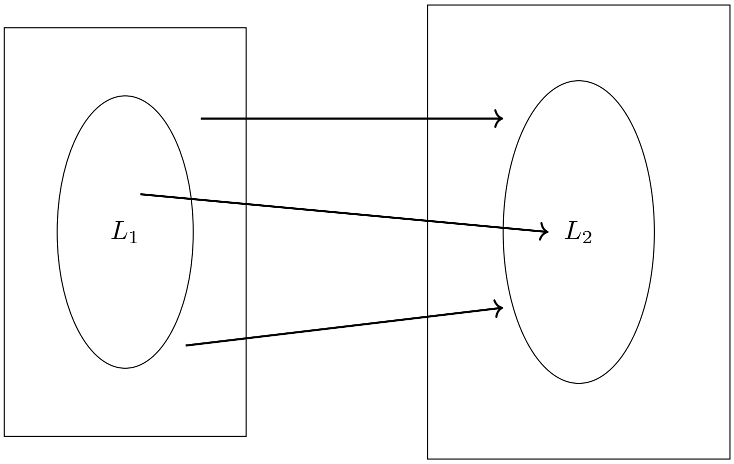
Let $L_1 \leq L_2$ *if* L_2 *is decidable, then* L_1 *is decidable.*

contrpositive: If L_1 *is undecidable, then* L_2 *is undecidable.*

M_2 *decides* L_2







If $x \in L_1$, then $M(x) \in L_2$, so M_2 accepts M(x)

If $x \notin L_1$, then $M(x) \notin L_2$, so M_2 rejects M(x)

***thm* 4.5.** *L_{halt} is undecidable*

Proof. *we will prove $L_{flip} \leq L'_{halt}$*

Assuming L_{halt} is decidable by a TM M_{halt}, we will prove L_{flip} is decidable, which would be a contradiction.

Create a TM M_{flip} as follows:

run M_{halt} on input (α, α)

1. *If M_{halt} rejects (α, α), let M_{flip} accept α*

2. *If M_{halt} accepts α, α, simulate M_α on input α (using a UTM), and flip the output*

It is easy to verify M_{flip} decides L_{flip}. Contradiction! □

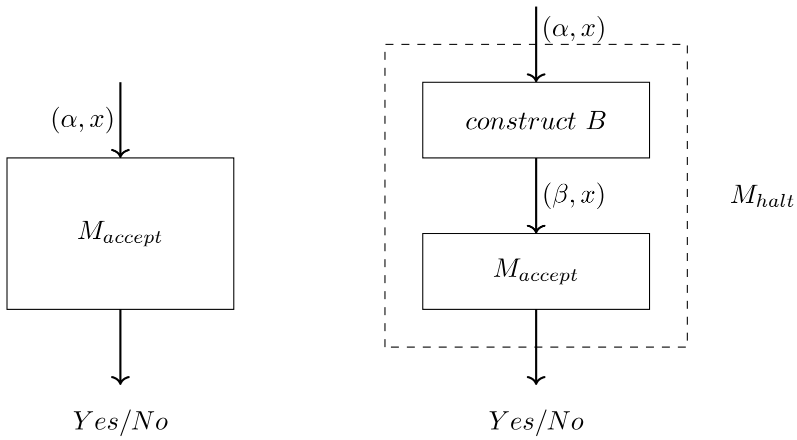
***lemma* 4.6.** *$L_{accept} = \{(\alpha, x), M_\alpha$ accepts $x\}$ is undecidable*

Proof. *we will prove $L_{halt} \leq L_{accept}$. Assuming for contradiction that L_{accept} is decidable, i.e. there exists a TM M_{accept} that decides L_{accept}, we construct a TM M_{halt} that decides L_{halt} as follows:*

1. *On input (α, x), create a new TM M_β, which simulates M_α on input x, and always accepts whenever M_alpha halts (If M_α loops forever, M_β loops forever as well)*

2. *Run M_{accept} on input (β, x), and forward its output. Clearly, M_{halt} decides L_{halt}. Contradiction!*

□



decides $if(\alpha, x) \in L_{accept}$

lemma 4.7. *Let* $L_{empty} = \{< M >$, *M does not accept any input, i.e.* $L(M) = \emptyset\}$

Proof. We will prove $L_{halt} \leq L_{empty}$. *Assuming for contradiction that* L_{empty} *can be decided by a TM* M_{empty}, *we construct a TM* M_{halt} *as follows:*
On input (α, x)

1. *We construct a new TM* M_β, *whose input is* $y \in \{0,1\}^*$, *as follows*

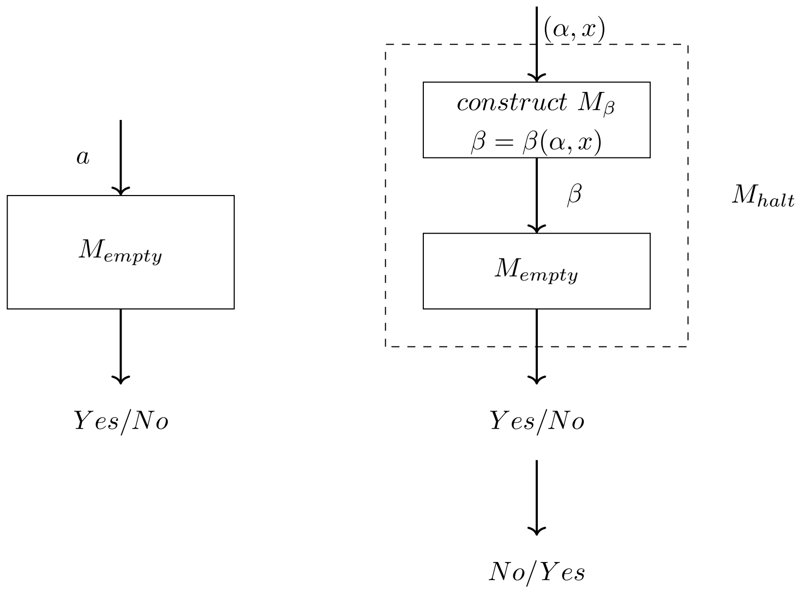
   (a) *simulate* M_alpha *on input* x

   (b) *if step (b) halts, always accept* y

   Clearly, $L(M_\beta) = \emptyset$ *if* M_α *does not halt on x. Otherwise,* $L(M_\beta) = \{0,1\}^*$

2. *Run* M_{empty} *on input* β *and flip the output. We can verify that* M_{halt} *decides* L_{halt}. *Contradiction!*

□



***thm* 4.8.** *Let $L_{regular} = \{< M >$, M is a TM, s.t. L(M) is a regular language$\}$, it is decidable*

Proof. *Assume for contradiction that $L_{regular}$ is decidable, i.e. $\exists$ a TM $M_{regular}$ that decides $L_{regular}$. We will prove L_{accept} is decidable.*

On input (α, x), construct a TM as follows:

1. *Construct a TM M_β, where $\beta = \beta(\alpha, x)$, and the input of M_β is denoted by y*
   (a) *If $y \in \{0^n 1^n, n \geq 0\}$, accept*
   (b) *Otherwise, simulate M_α on x, and accept iff M_α accepts x.*
2. *Run $M_{regular}$ on β, and forward its output*
   - *case1 $(\alpha, x) \notin L_{accept}$, i.e., M_{alpha} does not accept x*
     So, $L(M_\beta) = \{0^n 1^n, n \geq 0\}$, which is not a regular language
     Thus, $M_{regular}$ rejects β, which implies that M_{halt} rejects β
   - *case2 $(\alpha, x) \in L_{accept}$. So $L(M_\beta) = \{0, 1\}^*$, which is regular*
     As such, $M_{regular}$ accepts β, and so does M_{accept}

□

***lemma* 4.9.** *Let $L_{equal} = \{(< M_1 >, < M_2 >)$, M_1, M_2 are TMs, s.t. $L(M_1) = L(M_2)\}$, it's undecidable*

Proof. *Assuming for contradiction that L_{equal} is decidable, L_{equal} is decided by a TM M_{equal}*

On input $< M >$ construct a TM M_{empty} as follows

1. *Run M_{equal} on input $(< M >, < M_0 >)$, where M_0 rejects immediately $(L(M_0) = \emptyset)$*

2. *Forward the above output*

$\square$