Lecture 42

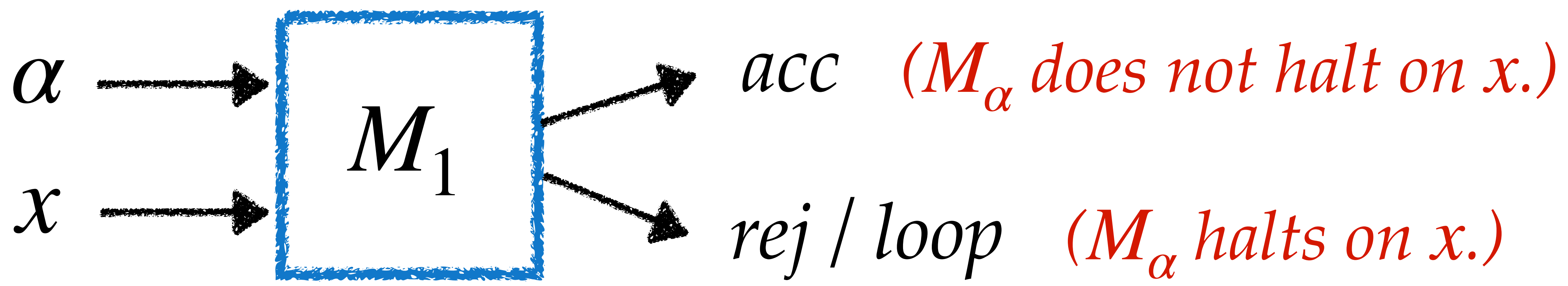
$\overline{HALT}$, More Undecidable Languages

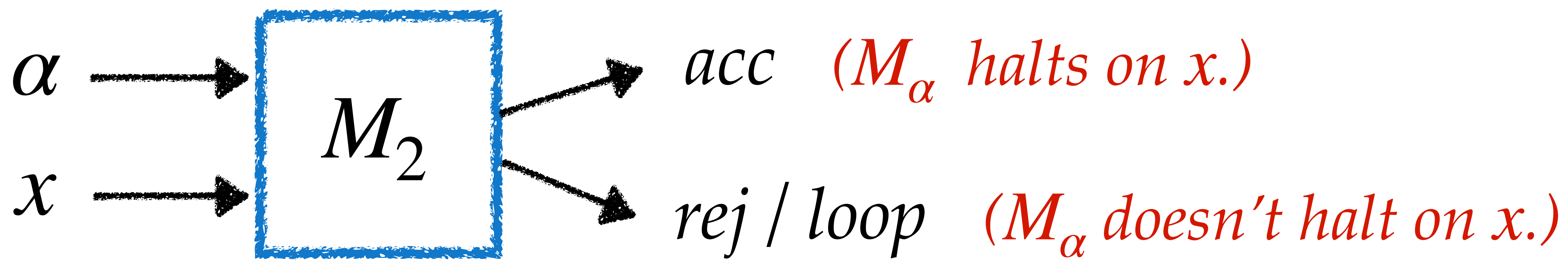
Unrecognisability of $\overline{HALT}$

Theorem: $\overline{HALT} = \{(\alpha, x) \mid M_\alpha$ does not halt on $x\}$ is unrecognisable.

Proof: Suppose $\overline{HALT}$ is recognisable and M_1 is a TM that recognises $\overline{HALT}$.

Let M_2 be a TM that recognises $HALT$.





Decider M for $HALT$ on input (α, x):

- Runs M_1 on (α, x) for one step , then M_2 on (α_2, x) for one step and so on.
- If M_1 accepts, then rejects.
- If M_2 accepts, then accepts.

A Few Observations

Theorem: If L is undecidable, at least one of L or $\overline{L}$ will be unrecognisable.

Proof Idea: The same as the proof of the last slide.

Theorem: If L is decidable, then $\overline{L}$ is decidable too.

Proof Idea: Swap the q_{accept} and q_{reject} in the transition function of decider of L.

Undecidability of $\mathbf{A}_{TM}$

Theorem: $A_{TM} = \{(\alpha, x) \mid M_\alpha \text{ accepts } x\}$ is undecidable.

Proof: Suppose A_{TM} has a decider M.

We can construct a decider M' for $HALT$ that on input (α, x):

- Construct α' from α by replacing every occurrence of q_{reject} by q_{accept}.
- Runs M on (α', x):
  - If M accepts (α', x), M' also accepts (α, x).
  - If M rejects (α', x), M' also rejects (α, x).

M_α halts on $x \implies M_{\alpha'}$ accepts x

M_α does not halt on $x \implies M_{\alpha'}$ does not accept x

■

Undecidability of Reg$_{TM}$

Theorem: $REG_{TM} = \{\alpha \mid L(M_\alpha) \text{ is regular}\}$ is undecidable.

Proof: Suppose REG_{TM} has a decider M.

We can construct a decider M' for $HALT$ that on input (α, x):

- Constructs a TM N that on input y does the following:
  - If y is a prime, it accepts it.
  - If y is a not a prime, starts running M_α on x
    - If M_α halts on x, then N will accept y.
    - If M_α does not halt on x, then N will also not halt.
- Runs M on $\langle N \rangle$:
  - If M accepts $\langle N \rangle$, M' also accepts (α, x).
  - If M rejects $\langle N \rangle$, M' also rejects (α, x).

■