5CCS2FC2: Foundations of Computing II

(Beyond) The Limits of Computation

Week 11

Dr Christopher Hampson

Department of Informatics

King's College London

Undecidability via Mapping Reductions

Mapping Reductions

Mapping Reductions

- **Mapping Reduction**
  - A **mapping reduction** from a problem X to a problem B is a **computable function** $f : \Sigma^* \to \Sigma^*$ such that

$$w \in A \iff f(w) \in B$$

$A \leq_m B$ means A is **(mapping) reducible** to B

Mapping Reductions

Theorem If $A \leq_m B$ and B is decidable, then A is also decidable.

Proof:

Step 1) Suppose that:

(i) $A \leq_m B$, and **(ii)** B is decidable

Mapping Reductions

Step 2) From **(i)** there is some **computable function** f such that

$$\boxed{w \in A \iff f(w) \in B}$$

Step 3) From **(ii)** there is some **sound**, **complete**, and **terminating** machine M_B such that

$$\boxed{u \in B \iff M_B(u) = 1}$$

Mapping Reductions

Step 4) Hence, we can construct a sound, complete, and terminating algorithm for A

$$\boxed{M_A(w) = M_B\left(f(w)\right)}$$

it follows that

$$\boxed{w \in A \iff f(w) \in B \iff M_B\left(f(w)\right) = 1 \iff M_A(w) = 1}$$

Q.E.D.

The Accepting Problem

Mapping Reductions

Theorem The Accepting Problem $\mathbf{A}_{TM}$ is Undecidable.

Proof:

Step 1) Since $\mathbf{HALT}_{TM}$ is **undecidable**, it is sufficient to show that

$$\boxed{\mathbf{HALT}_{TM} \ \leq_m \ \mathbf{A}_{TM}}$$

Mapping Reductions

Theorem The Accepting Problem $\mathbf{A}_{TM}$ is Undecidable.

Proof:

Step 1) Since $\mathbf{HALT}_{TM}$ is undecidable, it is sufficient to show that

$$\boxed{u \in \mathbf{HALT}_{TM} \quad \Longleftrightarrow \quad f(u) \in \mathbf{A}_{TM}}$$

(for some computable function $f : \Sigma^* \to \Sigma^*$)

Mapping Reductions

Theorem The Accepting Problem $\mathbf{A}_{TM}$ is Undecidable.

Proof:

Step 1) Since $\mathbf{HALT}_{TM}$ is **undecidable**, it is sufficient to show that

$$\langle \mathbf{code}(M), w \rangle \in \mathbf{HALT}_{TM} \iff f\left(\langle \mathbf{code}(M), w \rangle \right) \in \mathbf{A}_{TM}$$

(for some computable function $f : \Sigma^* \to \Sigma^*$)

Mapping Reductions

Theorem The Accepting Problem $\mathbf{A}_{TM}$ is Undecidable.

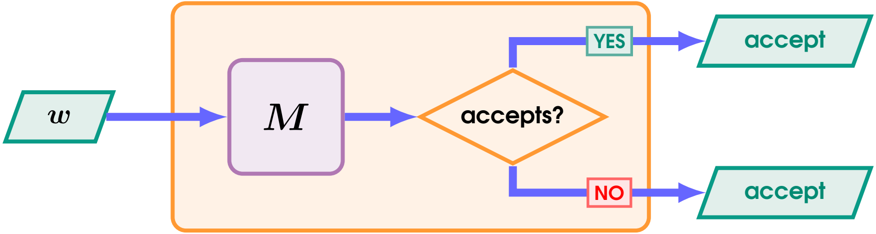
Proof:

Step 1) Since $\mathbf{HALT}_{TM}$ is **undecidable**, it is sufficient to show that

$$\boxed{\langle \textbf{code}(M), w\rangle \in \mathbf{HALT}_{TM} \quad \Longleftrightarrow \quad \langle \textbf{code}(M'), w'\rangle \in \mathbf{A}_{TM}}$$

Mapping Reductions

Step 2) Given any input machine M, we construct a **new machine** M' as follows:



$$M'(w) = 1 \iff M(w) = 1 \text{ OR } M(w) = 0$$

Mapping Reductions

Step 3) Hence we have that

$$\langle \mathbf{code}(M), w \rangle \in \mathbf{HALT}_{TM} \quad \Longleftrightarrow \quad \langle \mathbf{code}(M'), w \rangle \in \mathbf{A}_{TM}$$

- Hence, our **computable function** for our reduction is given by

$$f\left(\langle \mathbf{code}(M), w \rangle \right) \; = \; \langle \mathbf{code}(M'), w \rangle$$

Q.E.D.

Mapping Reductions

Step 3) Hence we have that

$$\langle \textbf{code}(M), w\rangle \in \textbf{HALT}_{TM} \iff \langle \textbf{code}(M'), w\rangle \in \textbf{A}_{TM}$$

- Hence, our **computable function** for our reduction is given by

$$f(u) = \begin{cases} \langle \textbf{code}(M'), w\rangle & \text{if } u = \langle \textbf{code}(M), w\rangle \\ \varepsilon & \text{otherwise} \end{cases}$$

Q.E.D.

The non-Emptiness Problem

Mapping Reductions

The Emptiness Problem E_{TM}

Input) An encoding of a Machine $\textbf{code}(M)$,

Output) **True** if and only if $\textbf{Language}\,(M) = \emptyset$.

$$\mathsf{E}_{TM} \;=\; \left\{ \textbf{code}(M) \;:\; \textbf{Language}\,(M) = \emptyset \right\}$$

Mapping Reductions

The *non*-Emptiness Problem $\overline{\mathsf{E}_{TM}}$

Input) An encoding of a Machine $\mathbf{code}(M)$,

Output) True if and only if $\mathbf{Language}\,(M) \neq \emptyset$.

$$\overline{\mathsf{E}_{TM}} \;=\; \left\{\mathbf{code}(M) \;:\; \mathbf{Language}\,(M) \neq \emptyset \;\right\}$$

Mapping Reductions

Theorem The *non*-Emptiness Problem $\overline{\mathbf{E}_{TM}}$ is Undecidable.

Proof:

Step 1) Since $\mathbf{A}_{TM}$ is **undecidable**, it is sufficient to show that

$$\boxed{\mathbf{A}_{TM} \leq_m \overline{\mathbf{E}_{TM}}}$$

Mapping Reductions

Theorem The *non*-Emptiness Problem $\overline{\mathbf{E}_{TM}}$ is Undecidable.

Proof:

Step 1) Since $\mathbf{A}_{TM}$ is **undecidable**, it is sufficient to show that

$$u \in \mathbf{A}_{TM} \quad \Longleftrightarrow \quad f(u) \in \overline{\mathbf{E}_{TM}}$$

(for some computable function $f : \Sigma^* \to \Sigma^*$)

Mapping Reductions

Theorem The *non*-Emptiness Problem $\overline{\mathbf{E}_{TM}}$ is Undecidable.

Proof:

Step 1) Since $\mathbf{A}_{TM}$ is **undecidable**, it is sufficient to show that

$$\langle \mathbf{code}(M), w \rangle \in \mathbf{A}_{TM} \iff f\left(\langle \mathbf{code}(M), w \rangle\right) \in \overline{\mathbf{E}_{TM}}$$

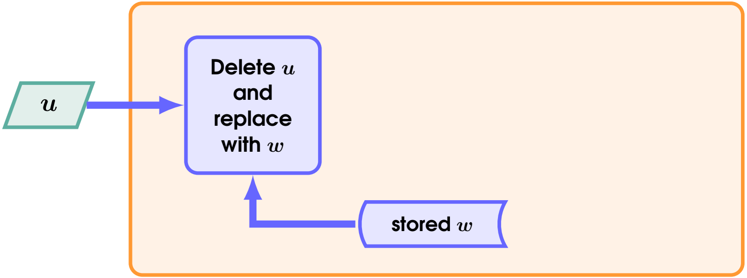
(for some computable function $f : \Sigma^* \to \Sigma^*$)

Mapping Reductions

Step 2) Given any input machine M and input word w, we construct a **new machine M_w** as follows:

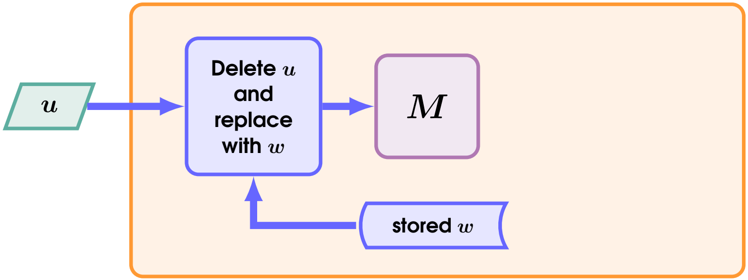
Mapping Reductions

Step 2) Given any input machine M and input word w, we construct a **new machine** M_w as follows:



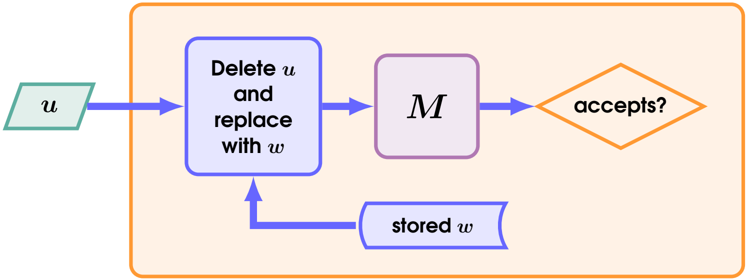
Mapping Reductions

Step 2) Given any input machine M and input word w, we construct a **new machine** M_w as follows:



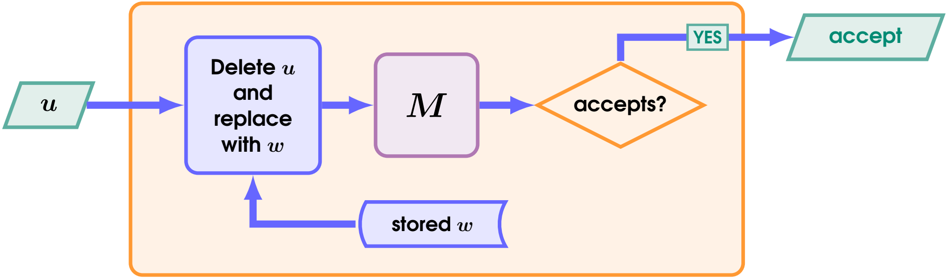
Mapping Reductions

Step 2) Given any input machine M and input word w, we construct a **new machine** M_w as follows:



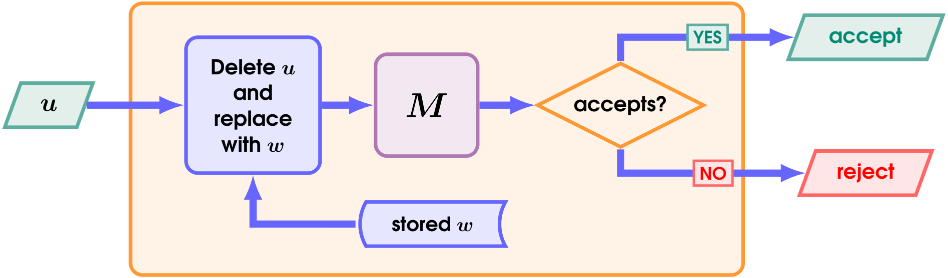
Mapping Reductions

Step 2) Given any input machine M and input word w, we construct a **new machine** M_w as follows:



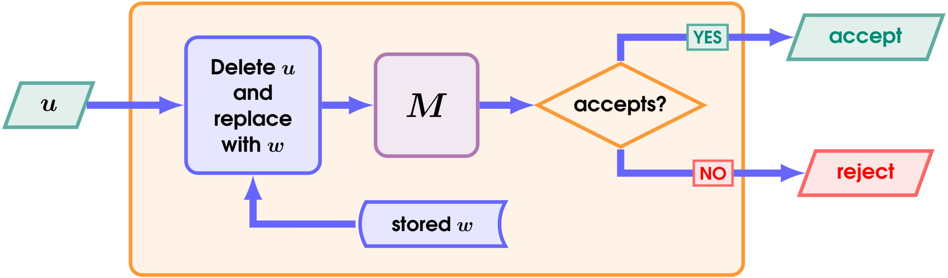
Mapping Reductions

Step 2) Given any input machine M and input word w, we construct a **new machine** M_w as follows:



Mapping Reductions

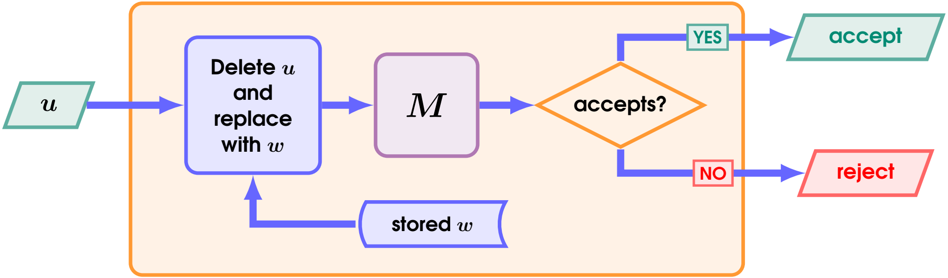
Step 2) Given any input machine M and input word w, we construct a **new machine** M_w as follows:



$$M_w(u) = 1 \iff M(w) = 1$$

Mapping Reductions

Step 2) Given any input machine M and input word w, we construct a **new machine** M_w as follows:



$$\textbf{Language}\,(M_w) \neq \emptyset \quad \Longleftrightarrow \quad M(w) = 1$$

Mapping Reductions

Step 3) Hence we have that

$$\langle \textbf{code}(M), w \rangle \in \textbf{A}_{TM} \quad \Longleftrightarrow \quad \textbf{code}(M_w) \in \overline{\textbf{E}_{TM}}$$

- Hence, our **computable function** for our reduction is given by

$$f\left(\langle \textbf{code}(M), w \rangle \right) = \textbf{code}(M_w)$$

Q.E.D.

Mapping Reductions

Step 3) Hence we have that

$$\langle \mathbf{code}(M), w \rangle \in \mathbf{A}_{TM} \quad \Longleftrightarrow \quad \mathbf{code}(M_w) \in \overline{\mathbf{E}_{TM}}$$

- Hence, our **computable function** for our reduction is given by

$$f(u) = \begin{cases} \mathbf{code}(M_w) & \text{if } u = \langle \mathbf{code}(M), w \rangle \\ \varepsilon & \text{otherwise} \end{cases}$$

Q.E.D.

End of Slides!

🎉

Mapping Reductions

Theorem If $A \leq_m B$ and A is undecidable, then B is undecidable.