COMP0017 Computability and Complexity Theory

Fabio Zanasi
http://www0.cs.ucl.ac.uk/staff/F.Zanasi/

Lecture nine

Previously on COMP0017

We discussed the concept of a **universal Turing machine** (UTM) and constructed one.

A key idea of the UTM is that a Turing machine may take other Turing machines as input data.

We used these insights to show that there exists an **undecidable** problem (the Halting problem) and an **unrecognisable** problem (the complement of the Halting problem).

One more example

We give one more example of an undecidable problem.

The *empty tape halting problem* (*ETH*) is defined by the following language.

$$ETH = \{\ x \in \Sigma^* \mid x = \text{code}(\mathcal{M}) \text{ and } \mathcal{M} \text{ halts on } \varepsilon.\}$$

Theorem *ETH* is undecidable.

ETH is undecidable

Proof We reason by contradiction. Suppose *ETH* was decidable, say by a TM $\mathcal{M}_{ETH}$. We now show how we can use $\mathcal{M}_{ETH}$ to construct a TM $\mathcal{M}_H$ deciding *HALT.*

$\mathcal{M}_H$ is defined as follows.

On input $\langle y,x \rangle$:

- If $y \neq code(\mathcal{M})$ for all $\mathcal{M}$, reject.
- If $y = code(\mathcal{M})$, construct $\mathcal{M}_{\mathcal{M},x}$ as follows:
  1. $\mathcal{M}_{\mathcal{M},x}$ enters a loop on any non-empty string.
  2. On input ε, write x on tape and simulate $\mathcal{M}$ on x.
- Accept if $\mathcal{M}_{ETH}$ accepts $code(\mathcal{M}_{\mathcal{M},x})$, otherwise reject.

ETH is undecidable

We conclude the proof by verifying that $\mathcal{M}_H$ decides the Halting problem.

$$\langle y, x \rangle \in HALT \iff y = \text{code}(\mathcal{M}) \text{ and } \mathcal{M} \text{ halts on x.} \iff \mathcal{M}_{\mathcal{M},x} \text{ halts on } \varepsilon. \iff \mathcal{M}_{ETH} \text{ accepts code}(\mathcal{M}_{\mathcal{M},x}).$$

$$\Updownarrow$$

$$\mathcal{M}_H \text{ accepts } \langle y, x \rangle.$$

But we know that the Halting problem is undecidable, contradiction. So, the machine $\mathcal{M}_{ETH}$ that we used to construct $\mathcal{M}_H$, cannot exist, meaning that *ETH* is undecidable.

A pattern emerges

Last lecture we saw:

Theorem $HALT^-$ is unrecognisable.

Proof If $HALT^-$ was recognisable, *HALT* would be decidable.

This lecture we saw:

Theorem *ETH* is undecidable.

Proof If *ETH* was decidable, *HALT* would be decidable.

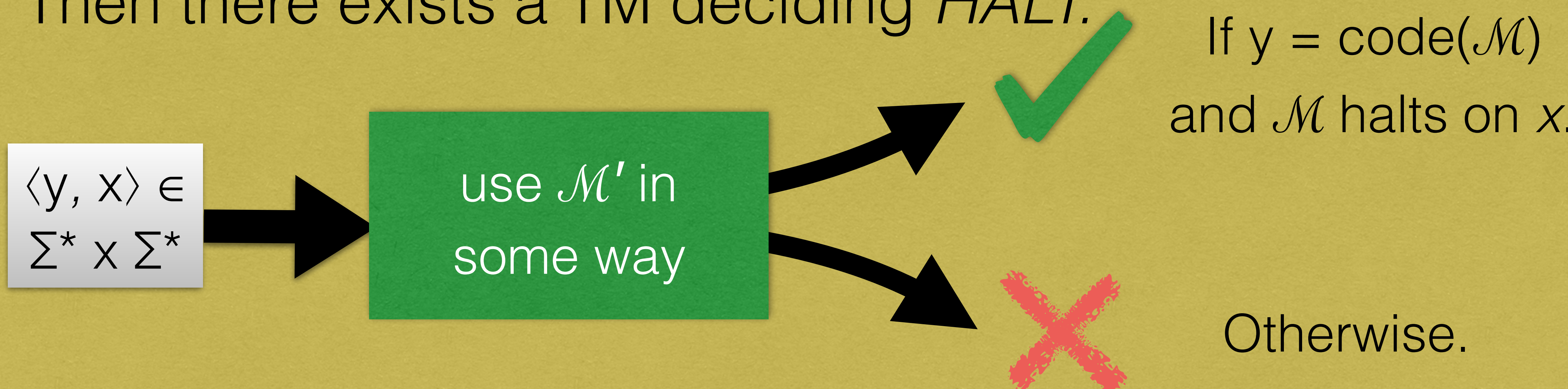
A pattern emerges

In each case, we reduce the decidability/recognisability of a problem *L* to the decidability of another problem (*HALT*).

$$HALT = \{ \langle y, x \rangle \in \Sigma^* \text{x}\ \Sigma^* \mid y = \text{code}(\mathcal{M}) \text{ and } \mathcal{M} \text{ halts on } x.\}$$

Proof outline: suppose a TM $\mathcal{M}'$ deciding *L* exists.

Then there exists a TM deciding *HALT*:



In more abstract terms

In order to prove that a problem *L* is undecidable:

1. Start with a problem *L'* that you know is undecidable.
2. Show that if you could decide *L* then you could decide *L'*.
3. Conclude that you cannot decide *L*.

(The same argument works with ``recognise'' in place of decide.)

The key step is 2: showing that deciding *L'* **reduces** to deciding *L*.

In the next lecture we will turn this kind of reduction argument into a proof strategy, which shows undecidability of problems much more straightforwardly.