

# Decision problems

---

ian.mcloughlin@gmit.ie

## Decision problem

$$f : S \rightarrow T \text{ where } |T| = 2$$

A decision problem is a map to a set with two elements.

### Example

$$f : \{0, 1\}^* \rightarrow \{0, 1\}$$

$$f(s) = 0 \Leftrightarrow |s| \equiv_2 0$$

### Another example

$$f : \{0, 1\}^* \rightarrow \{0, 1\}$$

$$f(s) = 0 \Leftrightarrow wt(s) \equiv_2 0$$

## Recap on Languages

**Alphabet:** finite set of symbols, denoted  $\Sigma$ .

**String:** tuple  $w$  over  $\Sigma$ .

**Star:** all strings over  $\Sigma$ , denoted  $\Sigma^*$ .

**Language:** subset  $L$  of  $\Sigma^*$ .

**Length:** of a string, denoted  $|w|$ .

## Turing machines recap

For a given input a Turing machine does one of three things:

**Accepts** the input string by finishing in the accept state in a finite number of steps.

**Rejects** the input string by finishing in the reject/fail state in a finite number of steps.

**Continues** indefinitely in some sort of infinite loop.

Remember there are a finite number of states and tape symbols.

# Deciders

$$f : \Sigma^* \rightarrow \{q_f, q_a\}$$

**Decider:** a Turing machine that always finishes in a finite number of steps.

**Decides:** decides the language it accepts.

**Decidable:** a language is called decidable if any Turing machine decides it.

Important question: are all languages decidable?

# PRIMES

Is there a decider that decides if the input is a prime number?

Yes

We can make a Turing machine that counts up to a number, divides and checks the remainder is zero.

We call the language containing strings representing the prime numbers in binary form PRIMES.

$$\text{PRIMES} = \{x \mid x \in \mathbb{N} \text{ and } x \text{ is prime}\}$$

So, PRIMES is decidable.

## An undecidable language

- Encode all Turing machines as strings over some finite alphabet.
- Consider the subset of Turing machines that are deciders.
- This set is undecidable.