Universal language

- $\langle M, w \rangle$: encoding $\langle M \rangle$ of M concatenated with $w \in \{0,1\}^*$.
- *Universal language*
$$L_u = \{\langle M, w \rangle \mid M \text{ accepts } w\}$$
- **Theorem.** L_u is recursively enumerable.
- A Turing machine U accepting L_u is called *universal Turing machine.*
- **Theorem** (Turing 1936). L_u is not recursive.

Decision problems

- Decision problems are problems with answer either yes or no.
- Associate with a language $L \subseteq \Sigma^*$ the decision problem D_L

  Input: $w \in \Sigma^*$

  Output: $\begin{cases} \text{yes}, & \text{if } w \in L \\ \text{no}, & \text{if } w \notin L \end{cases}$

  and vice versa.
- D_L is *decidable* (resp. *semi-decidable*) if L is recursive (resp. recursively enumerable).
- D_L is *undecidable* if L is not recursive.

Reductions

- A *many-one reduction* of $L_1 \subseteq \Sigma_1^*$ to $L_2 \subseteq \Sigma_2^*$ is a computable function $f : \Sigma_1^* \to \Sigma_2^*$ with $w \in L_1 \Leftrightarrow f(w) \in L_2$.
- **Proposition.** If L_1 is many-one reducible to L_2, then
  1. L_1 is decidable if L_2 is decidable.
  2. L_2 is undecidable if L_1 is undecidable.

Post's correspondence problem

- Given pairs of words
$$(v_1, w_1), (v_2, w_2), \dots, (v_k, w_k)$$
  over an alphabet Σ, does there exist a sequence of integers $i_1, \dots, i_m, m \geq 1$, such that
$$v_{i_1}, \dots, v_{i_m} = w_{i_1}, \dots, w_{i_m}.$$
- *Example*

i	v_i	w_i
1	1	111
2	10111	10
3	10	0

$$\Rightarrow v_2 v_1 v_1 v_3 = w_2 w_1 w_1 w_3 = 101111110$$

- **Theorem** (Post 1946). Post's correspondence problem is undecidable.

Hilbert's Tenth Problem

Hilbert, International Congress of Mathematicians, Paris, 1900

Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.

Theorem (Matiyasevich 1970)
Hilbert's tenth problem is undecidable.

Non-deterministic Turing machines

- *Next move relation:*
$$\delta \subseteq (Q \times \Gamma) \times (Q \times \Gamma \times \{L, R\})$$
- $L(M)$ = set of words $w \in \Sigma^*$ for which *there exists* a sequence of moves accepting w.
- **Proposition.** If L is accepted by a non-deterministic Turing machine M_1, then L is accepted by some deterministic machine M_2.

Time complexity

- M a (deterministic) Turing machine that halts on all inputs.
- Time complexity function $T_M : \mathbb{N} \to \mathbb{N}$
$$\begin{aligned} T_M(n) \quad = \quad & \max\{m \mid \exists w \in \Sigma^*, |w| = n \text{ such that the computation} \\ & \text{of } M \text{ on } w \text{ takes } m \text{ moves}\} \end{aligned}$$
(assume numbers are coded in binary format)
- A Turing machine is *polynomial* if there exists a polynomial $p(n)$ with $T_M(n) \leq p(n)$, for all $n \in \mathbb{N}$.
- The *complexity class P* is the class of languages decided by a polynomial Turing machine.

Time complexity of non-deterministic Turing machines

- M non-deterministic Turing machine
- The running time of M on $w \in \Sigma^*$ is
  - the length of a shortest sequence of moves accepting w if $w \in L(M)$
  - 1, if $w \notin L(M)$
- $T_M(n) = \max\{m \mid \exists w \in \Sigma^*, |w| = n \text{ such that the running time of } M \text{ on } w \text{ is } m\}$
- The *complexity class NP* is the class of languages accepted by a polynomial non-deterministic Turing machine.