

# Ograniczenia turingowskiego, **cyfrowego**, modelu obliczeń

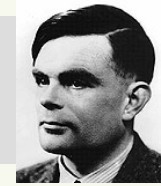
*Problemy nieobliczalne*

# *Co to znaczy „obliczać” (to compute)?*

- **OBLICZAĆ = PRZETWARZAĆ DANE**

- *przy założeniu jednak, że w matematycznej teorii danym odpowiadają pewne liczby, a programom – funkcje.*

# *Model obliczeń cyfrowych*



Najbardziej powszechny (być może: jedyny możliwy) model obliczeń wyznacza pojęcie **maszyny Turinga**.

Skończone realizacje programów dla **komputerów cyfrowych** są realnymi odpowiednikami obliczeń opisywanych w tym modelu.

Jest to model obliczeń **cyfrowych** (dyskretnych) i deterministycznych.



# *Co to znaczy „obliczać cyfrowo?”*

- **OBLICZAĆ = PRZETWARZAĆ DANE**

- *przy założeniu jednak, że w matematycznej teorii danym odpowiadają pewne liczby, a programom – funkcje.*

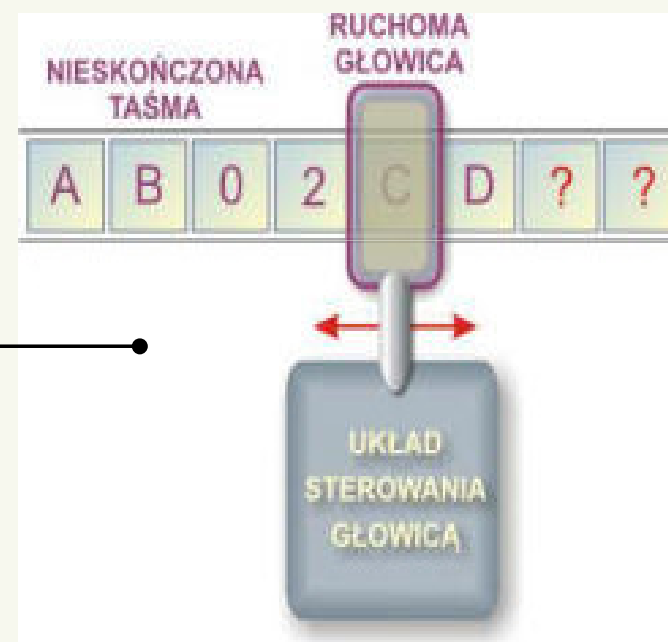
- ◆ **W **cyfrowym** modelu obliczeń, zdefiniowanym za pomocą pojęcia uniwersalnej **maszyny Turinga**:**

- *danym odpowiadają **liczby obliczalne** (w sensie Turinga),*
- *obliczeniom zaś **funkcje rekurencyjne**, czyli obliczalne w sensie Turinga.*

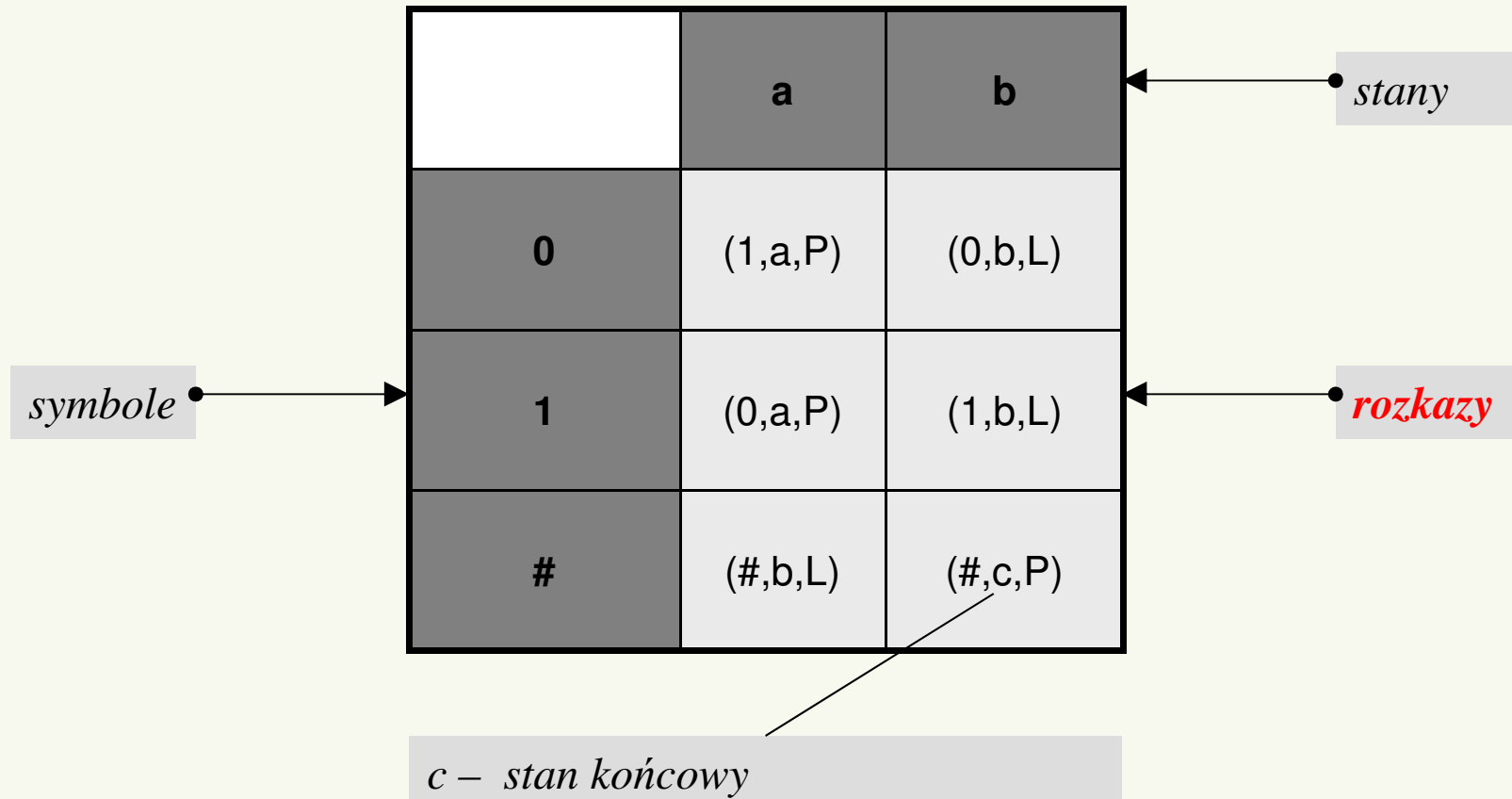
# *Jak jest „zbudowana” maszyna Turinga ?*

- **Maszyna Turinga** składa się z:
  - (1) nieskończonej, podzielonej na odrębne komórki, **taśmy**;
  - (2) **głowicy** do odczytu-zapisu danych;
  - (3) **rejestr** stanów;
  - (4) **tablicy** przejść między stanami.

Maszyna „działa” na podstawie programu zawartego w tablicy (4).

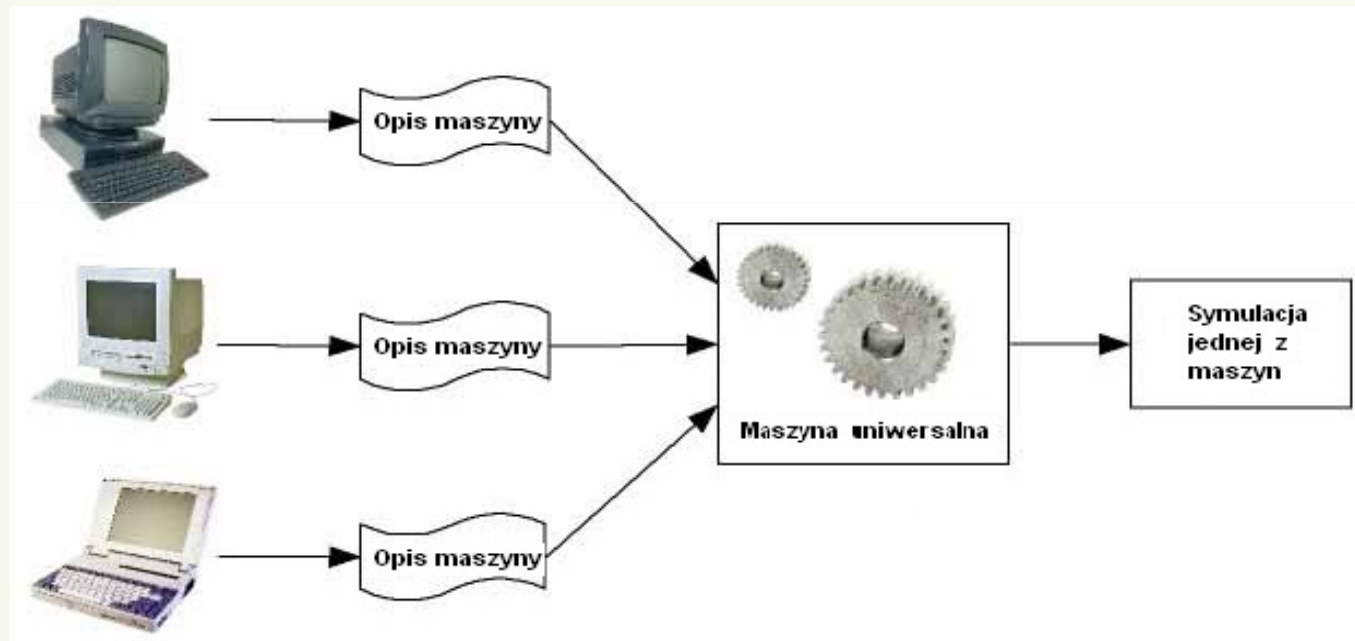


# *Jak wygląda program maszyny Turinga?*



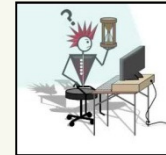
# Czym jest uniwersalna maszyna Turinga?

- **UMT** jest specjalną maszyną Turinga, której program ma za zadanie **symulować** działanie dowolnej, konkretnej MT.



- Wykazano, że UMT może wykonać dowolnie złożony program dla dowolnie zaawansowanej technicznie **maszyny cyfrowej**.

# *Na czym polegają ograniczenia modelu turingowskiego?*





# *Na czym polegają ograniczenia modelu turingowskiego?*

Polegają na tym, że **nie wszystkie problemy** mogą zostać rozwiązane za pomocą operacji określonych w tym modelu

O tym, że nie mogą, decydują dwie krytyczne własności algorytmów:

- a) **złożoność czasowa** (zbyt wysoka)
- b) **własność stopu** (gdy nie zachodzi)

Innymi słowy: **istnieją problemy nieobliczalne!**

# *Problemy nieobliczalne*

Problemy **nieobliczalne** (w modelu Turinga)  
dzielą się na:

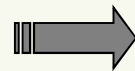
- a) nieobliczalne **praktycznie** – gdy dla danego problemu nie istnieje algorytm o dostatecznie niskiej złożoności czasowej,
- b) nieobliczalne **zasadniczo** – gdy nie istnieje algorytm rozwiązujący wszystkie przypadki szczególne danego problemu.



# Problemy o złożoności wykładniczej

## Problem spełnialności (logika)

- Czy istnieje takie wartościowanie zmiennych zdaniowych, przy którym formuła zawierająca  $n$  zmiennych jest prawdziwa?

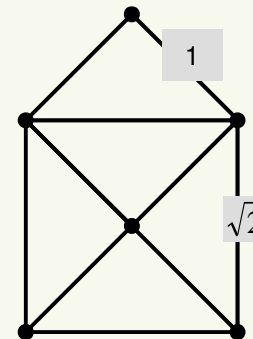


$$(\neg b \wedge (a \rightarrow b)) \rightarrow (\neg a)$$

1                      0                      1                      0

## Problem komiwojażera (grafy)

- W danym grafie z określonymi wagami krawędzi znajdź ścieżkę zawierającą wszystkie wierzchołki, która ma najmniejszą sumaryczną wagę (ścieżkę najkrótszą).



# Problemy o złożoności silniowej

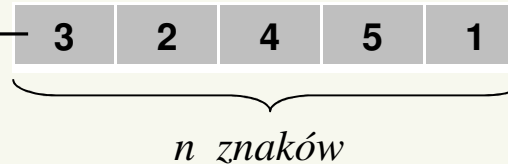
Złożoność czasową  $n!$  mają:

- algorytmy sprawdzające wszystkie permutacje danych wejściowych o rozmiarze  $n$ .

$n!$  to liczba permutacji zbioru  $n$ -elementowego.

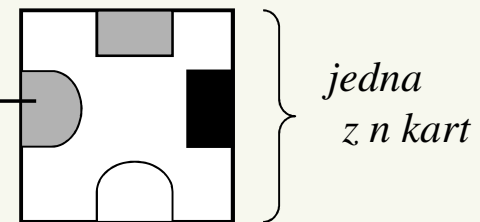
Przykład 1 (szyfr)

- Odgadnij szyfr  $n$ -znakowy złożony z  $n$  różnych znaków nie powtarzających się.



Przykład 2 (układanka)

- Czy dla danych  $n$  kart istnieje złożony z nich kwadrat, w którym wszystkie karty stykają się odpowiednimi bokami?



*Jaki problem nieobliczalny  
(zasadniczo) uznaje się za  
kanoniczny?*



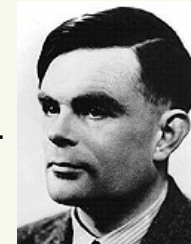
# *Turingowski problem stopu*

**Problem stopu** (maszyny Turinga)

✓ Dla dowolnej maszyny  $MT_i$  i jej dowolnych danych wejściowych  $D_j$

*odpowiedz jednoznacznie,*

czy  $MT_i$  zatrzyma się dla danych  $D_j$   
tj. zakończy przetwarzanie danych  $D_j$ ?



Mniej technicznie:

Czy istnieje taki uniwersalny algorytm, który analizując zapis każdego innego algorytmu oraz dowolnych jego danych,

*rozstrzygnie jednoznacznie*

czy analizowany algorytm zakończy przetwarzanie swoich danych, czy też będzie je przetwarzał w nieskończoność?

# *Inne pr. nieobliczalne zasadniczo*

Czy dane równanie *diofantyczne*, z dowolną liczbą niewiadomych i całkowitymi współczynnikami, ma choć jedno rozwiązanie w zbiorze liczb całkowitych ?

•  $x^2 + 2y^3 - 4y^2 + z^4 = 0$

Czy dane dwa języki sztuczne, z określonymi regułami budowania słów, pozwalają zbudować, zgodnie ze swoimi regułami, to samo (dane z góry i dowolne) *słowo* ?

• *abbbcaabbbbababc*

# *Problemy nieobliczalne - pytania*

Istnienie problemów nieobliczalnych (PNB) prowadzi do następujących **pytań**:

- a) Czy dla potrzeb realnych zastosowań nie wystarczą nam rozwiązania **problemów podobnych**, lecz obliczalnych?
- b) Czy z każdego z problemu PNB nie daje się wydzielić takich **podproblemów**, dla których istnieją efektywne algorytmy „lokalne”?
- c) Czy na gruncie **innych modeli obliczeń** niż cyfrowy (turingowski) problemy PNB nie stają się rozwiązywalne?



# *Obliczenia nieturingowskie*

Algorytmy wykraczające poza model Turinga (UMT) sytuują się w sferze tzw. **hiperobliczeń**, do których należą:

- obliczenia **analogowe** (ciągłe)
- obliczenia kwantowe
- obliczenia infinitystyczne

Mimo istnienia teoretycznych modeli różnych typów hiperobliczeń, wciąż rozważa się **pytania**: a) o ich relację do modelu Turinga, b) o ich fizyczną realizowalność.

