Informacja w ujęciu obliczeniowym?

Informacja w informatyce?

Dane?

Cztery „punkty" odniesienia …
(dla pojęcia informacji)

Czym jest informacja (ogólnie)?

COŚ w ŚWIECIE:

- organizacja, **forma**, sposób uporządkowania
  (*coś, co przeciwstawia się nieporządkowi i chaosowi*)

COŚ w LUDZKIM UMYŚLE:

- treść naszych **myśli**, czasem wiedza
  (*coś, co umysł przyswaja, przetwarza, przekazuje, wykorzystuje do kierowania ciałem…*)

COŚ w JĘZYKU:

- treść pewnych **symbolicznych zapisów**
  (*coś, co jest wyrażone w określonym kodzie; możliwym do zinterpretowania przez ludzki umysł*)

COŚ wewnątrz KOMPUTERA:

- informatyczny kod (przetwarzany lub sterujący), **dane**

Informacja jako coś w komputerze

- **Informacja = (komputerowe) dane**

  Informacja w kontekście informatycznym to tyle co **dane**, czyli odpowiednio ustrukturyzowane, zakodowane i fizycznie reprezentowane tworzywo maszyn informatycznych, np. cyfrowych.

Jest to pewien **informatyczny kod** (najczęściej możliwy do zinterpretowania przez człowieka)

Informatyczna triada

Za trzy najważniejsze pojęcia informatyki, określające (bardzo zgrubnie) jej specyfikę, trzeba uznać:

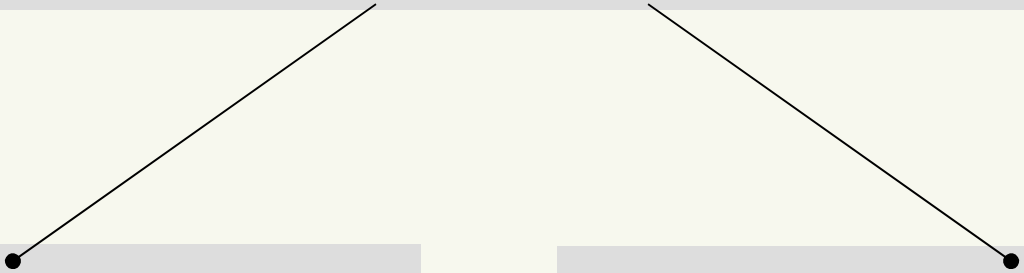
- *informację (dane), algorytm i automat.*

- **Informatyka** zajmuje się bowiem algorytmicznym przetwarzaniem informacji (ściślej: danych) za pomocą określonego rodzaju automatów (np. maszyn cyfrowych).

- **System informatyczny** jest pewnym automatem, który przetwarza dane w sposób algorytmiczny (czyli zaprogramowany).

Informacja, dane i liczby

Informacja > Dane > Liczby



- Informacja to coś więcej niż **dane**, bo dane stanowią pewną tylko, to znaczy komputerowo dostępną, formę informacji.

- Dane to coś więcej niż **liczby**, bo jakkolwiek dane koduje się liczbowo, to są one dodatkowo związane pewnymi informatycznymi strukturami.

Ale mimo wszystko: na poziomie najprostszego matematycznego opisu komputerowym danym odpowiadają jakieś liczby, a operacjom na danych jakieś **obliczenia**.

Co to znaczy „obliczać” (to compute)?

Co to znaczy „obliczać” (to compute)?

liczyć, rachować…
(wykonywać operacje na liczbach, znajdować wartości pewnych funkcji)

przetwarzać dane za pomocą komputerów…
(na podstawie programów, program > funkcja)

realizować algorytmy, czyli skończone zbiory reguł
(algorytmicznie rozwiązywać problemy)

działać zgodnie z pewnym modelem obliczeń
(np. cyfrowym $\equiv_{(np.)}$ uniwersalnej maszynie Turinga)

A zatem: co to znaczy „obliczać”?
- w sensie informatycznym -

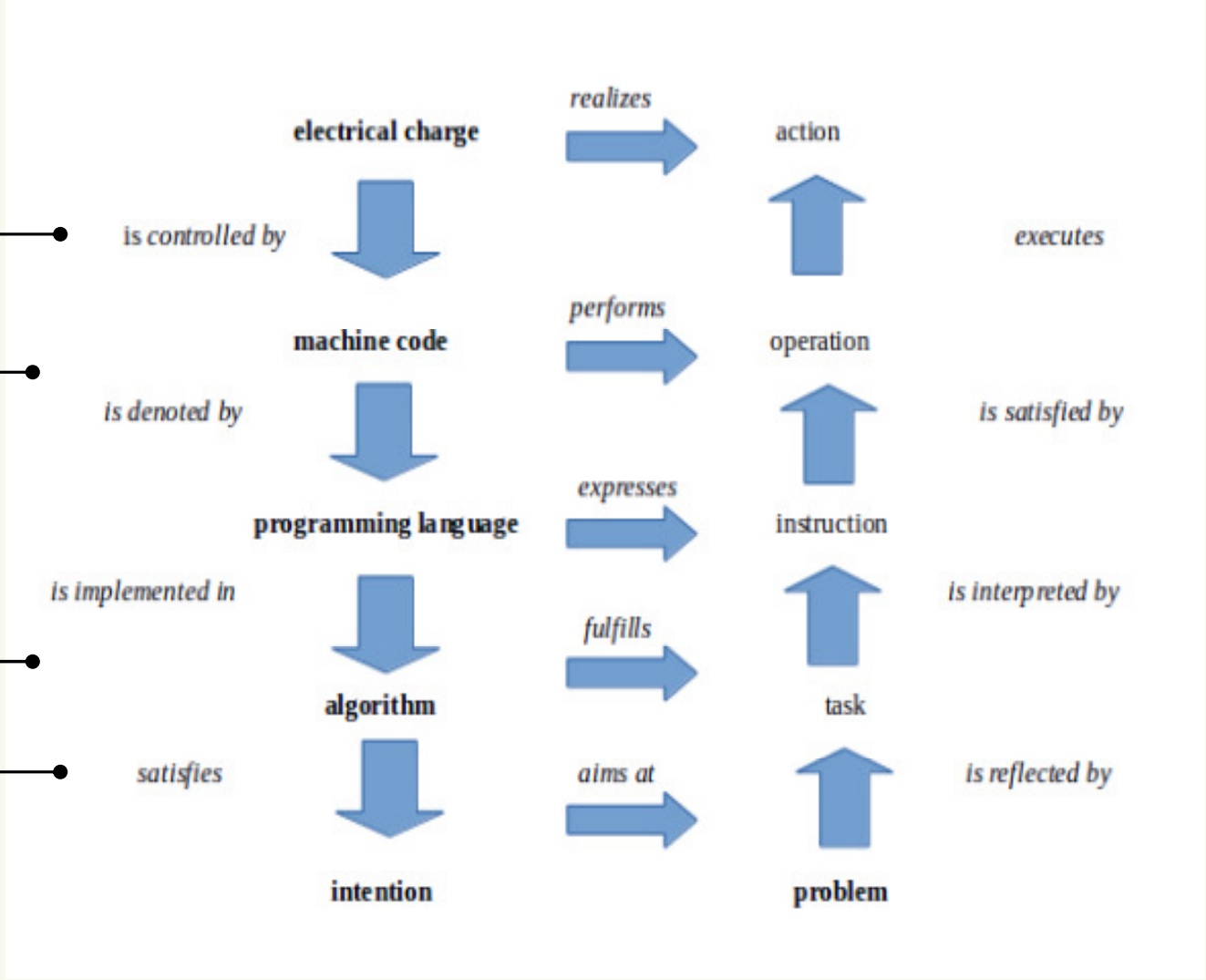
- **OBLICZAĆ = PRZETWARZAĆ DANE**
  - *przy założeniu jednak, że w matematycznej teorii danym odpowiadają pewne* ***liczby****, zaś algorytmom/programom –* ***funkcje****.*

Obliczenia: kolejne warstwy abstrakcji

warstwa **fizyczna**

warstwy **programistyczne**

warstwa **specyfikacji**



Dyskusja w blogu nad tekstem Giuseppe Primiero

Adres dyskusji: http://marciszewski.eu/?p=10493

- Na warstwie najniższej (u góry) informację utożsamia się z układem **fizycznych** (elektrycznych) stanów maszyny, które są kontrolowane za pomocą struktur określonych na wyższych warstwach.

(…)

- W moim odczuciu, w obrębie wszystkich warstw, informacja jest rozumiana jako **relacja** pomiędzy wejściem i wyjściem.

Z tego względu jest ona czymś **dynamicznym** (algorytmicznym, obliczeniowym…), jest wprawdzie pewną strukturą, ujmowaną na różnych poziomach w różny sposób, ale **strukturą relacyjną**, warunkującą działanie (action, operation, problem solving).

Zagadnienia otwarte

1) Czy tylko jedna warstwa, ta najniższa, jest **warstwą fizykalną**?

2) Czy wszystkie warstwy są równie istotne dla ujęcia istoty informacji/danych? Na przykład: czy trzech warstw wewnętrznych (kod maszynowy, program, algorytm) nie należałoby „zwinąć" do postaci jednej, **algorytmiczno-programistycznej**, warstwy?

3) Czy, a ewentualnie jak, należy uzupełnić podany schemat o warstwę **modelu obliczeń**?

4) Czy dobrze rozumiem, że przedstawiony schemat dotyczy informacji „jak" (jak z danych wejściowych otrzymać wyjściowe), dotyczy zatem **informacji sterującej** (działaniem maszyny)?
W ten sposób informację utożsamia się z algorytmem.

5) Jak rozumieć warstwę **intencji**? Czy chodzi o intencję „w czyjejś głowie"? Czy intencję/cel wskazaną formalnie (np. jako warunek rozwiązania pewnego problemu)?

Typy liczb a modele obliczeń

Każdy model obliczeń określa dozwolony typ danych, któremu to typowi odpowiada z kolei pewien rodzaj liczb.

Krótko: każdy model odwołuje się do pewnego typu liczb.

- obliczenia **cyfrowe** → liczby **naturalne** (wielkości dyskretne)
- obliczenia **analogowe** → liczby **rzeczywiste** (wielkości ciągłe)
- obliczenia **kwantowe** →? liczby **zespolone** (q-bity, stany kwantowe)

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.

$\approx$

Czym jest maszyna Turinga ?

- **Maszyna Turinga** składa się z*:*

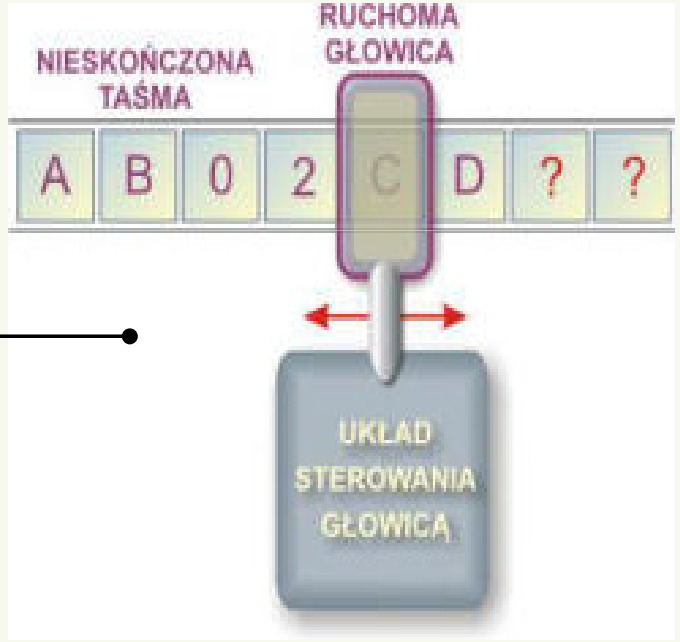
(1) nieskończonej, podzielonej na odrębne komórki, **taśmy**;

(2) **głowicy** do odczytu-zapisu danych;

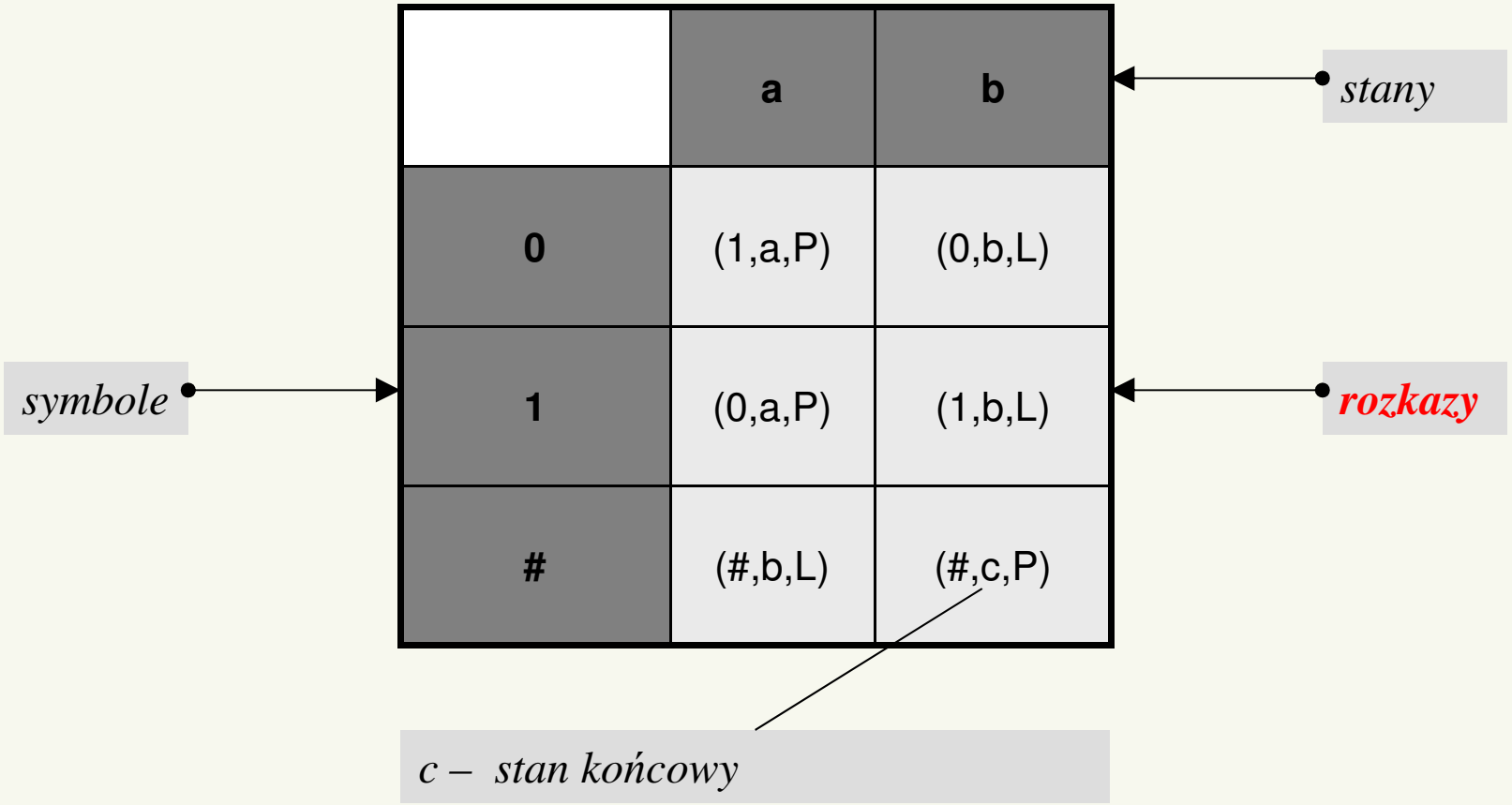
(3) **rejestru** stanów;

(4) **tablicy** przejść między stanami.

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

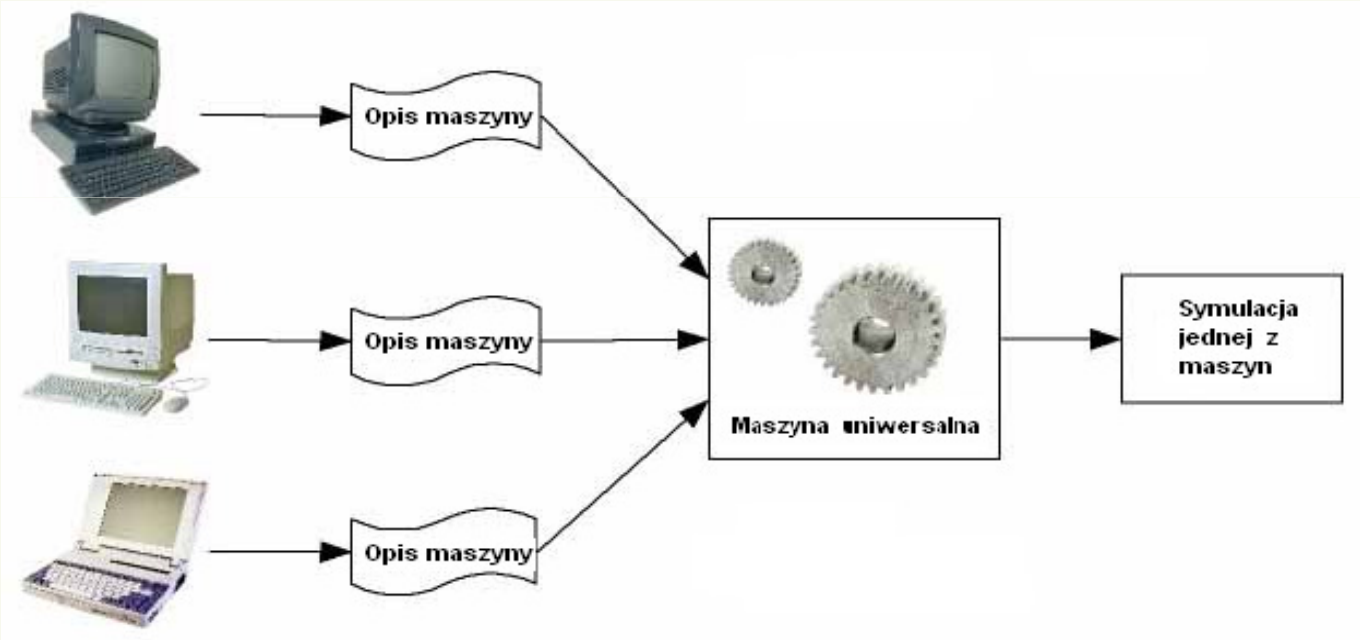


Jak wygląda program maszyny Turinga?



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**.

Istota cyfrowości

Najbardziej ogólna zasada cyfrowego przetwarzania danych jest następująca:

- zarówno same dane, jak i schematy ich przetwarzania (programy), mają postać kodu symbolicznego, złożonego z rozróżnialnych, dyskretnych elementów;

Ponadto:
kody symboliczne/cyfrowe mają dyskretną (de facto: skończoną) interpretację liczbową (w postaci najbardziej „oszczędnej” są to kody zero-jedynkowe).

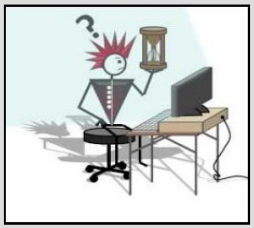
Zaskakujący wynik Turinga

- Istnieją **liczby**, których **nie można obliczyć**.

  (*w modelu cyfrowym*)

- Istnieją problemy (ściśle zdefiniowane), których **nie można rozwiązać**.

  (*w modelu cyfrowym*)

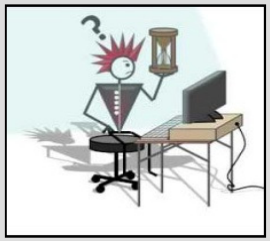


Problemy nieobliczalne

W turingowskim modelu obliczeń (cyfrowych) istnieją **problemy nieobliczalne**, czyli algorytmicznie nierozwiązywalne.

- Problemy te są:
  a) **nieobliczalne zasadniczo** – gdy nie istnieje jeden uniwersalny algorytm rozwiązujący wszystkie przypadki szczególne danego problemu,

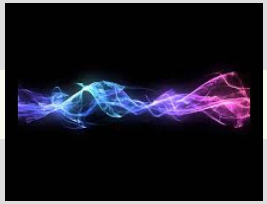
  b) **nieobliczalne praktycznie** – gdy dla danego problemu nie istnieje algorytm o dostatecznie niskiej złożoności obliczeniowej (np. czasowej).



Inne modele (hiper)obliczeń

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

- obliczenia **analogowe** (ciągłe)
- obliczenia losowe (np. kwantowe)
- obliczenia infinitystyczne



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

Jeszcze raz: co to znaczy „obliczać”?
- obliczenia a liczby -

- **OBLICZAĆ = PRZETWARZAĆ DANE**
  - *przy założeniu jednak, że w matematycznej teorii danym odpowiadają pewne **liczby**, zaś algorytmom/programom – **funkcje**.*

Typy liczb a modele obliczeń

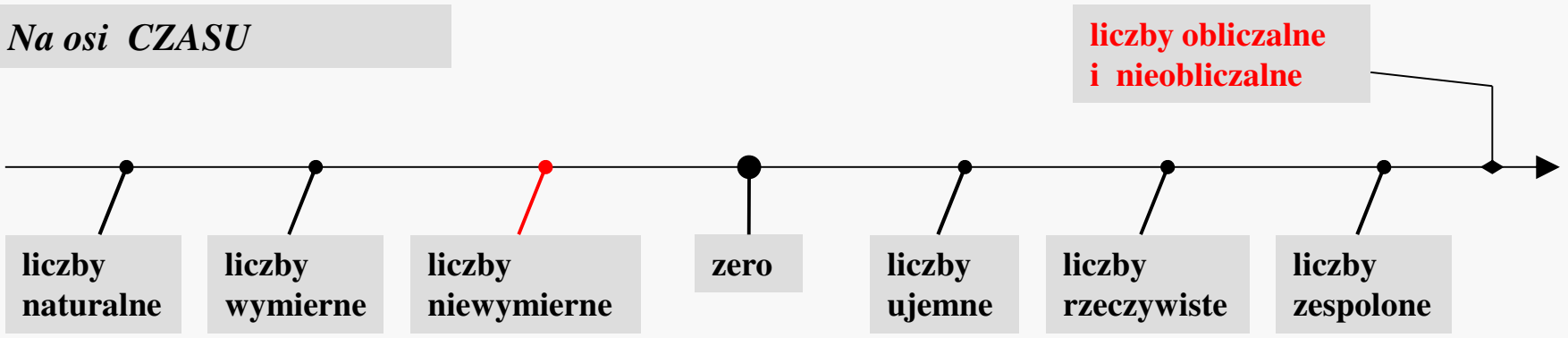
Każdy model obliczeń określa dozwolony typ danych, któremu to typowi odpowiada z kolei pewien rodzaj liczb.

Krótko: każdy model odwołuje się do pewnego typu liczb.

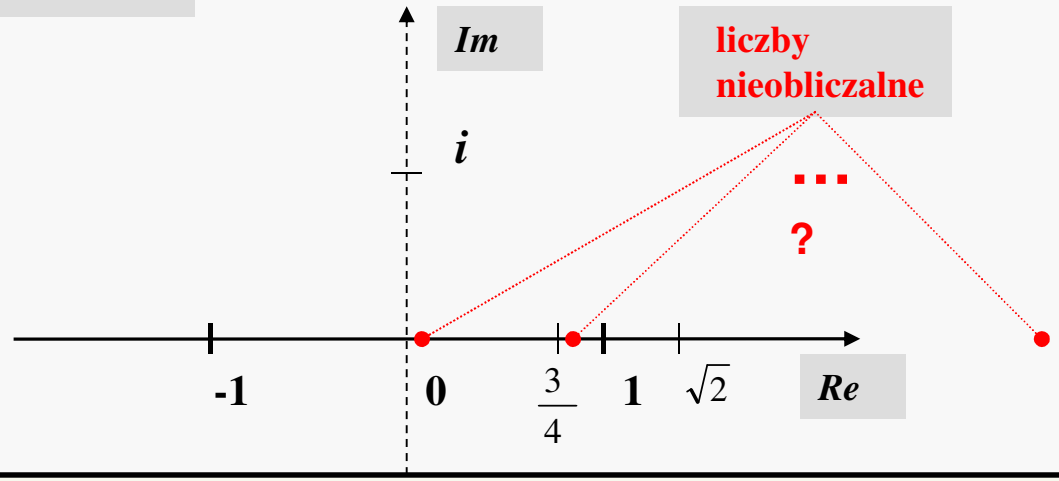
- obliczenia **cyfrowe** → liczby **naturalne** (wielkości dyskretne)
- obliczenia **analogowe** → liczby **rzeczywiste** (wielkości ciągłe)
- obliczenia **kwantowe** →? liczby **zespolone** (q-bity, stany kwantowe)

Różne klasy liczb

Na osi CZASU



Na osiach LICZBOWYCH



Liczby obliczalne i nieobliczalne

Liczba obliczalna jest to taka liczba rzeczywista, dla której istnieje **maszyna Turinga** (inaczej: program dla maszyny cyfrowej) pozwalająca obliczyć ją z dowolną zadaną dokładnością.

Liczba nieobliczalna nie ma powyższej własności: jest niewyznaczalna za pomocą maszyn Turinga.

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \ldots = 4 \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots = \sum_{n=0}^{\infty} \frac{1}{n!}$$

najszerzej znane obliczalne liczby niewymierne

Liczby nie(obliczalne) – *kilka określeń*

Liczby obliczalne…
(wg. Turinga)

- Istnieje klasyczny algorytm ich obliczania.
- Istnieje maszyna Turinga obliczająca je z dowolną zadaną dokładnością.
- Istnieje algorytm obliczania ich kolejnych cyfr.

Liczby nieobliczalne…
(wg. Turinga)

- Nie istnieje klasyczny algorytm ich obliczania.
- Żadna maszyna Turinga nie potrafi obliczyć ich z dowolną zadaną dokładnością.
- Nie istnieje algorytm obliczania ich kolejnych cyfr.

Definiowanie liczb nieobliczalnych

Niektóre liczby nieobliczalne można **zdefiniować**, nie podając jednak efektywnego schematu wyznaczania ich kolejnych cyfr.

Poglądowy przykład:

a) Tworzymy uporządkowaną listę maszyn Turinga MT_i (z danymi wejściowymi na taśmach).

b) Definiujemy liczbę zapisaną binarnie

L = $0,b_1b_2b_3b_4\ldots$ przy czym

- $b_i = 1$, gdy MT_i kończy pracę
- $b_i = 0$, gdy MT_i działa w nieskończoność