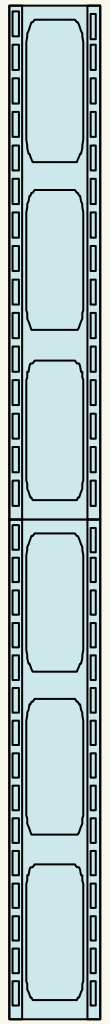
Informatyczne kłopoty z nieskończonością ?

czyli

O zdrowym sceptycyzmie w filozofii informatyki.

Struktura referatu

- Nieskończoność a sceptycyzm
- Nieskończoność w kontekście informatycznym
- Problemy nieobliczalne zasadniczo
- Problemy nieobliczalne praktycznie
- Nieskończony regres programów uczących się
- Nieskończony regres funkcji oceny w programach ewolucyjnych
- Pytania filozoficzne



Tropem dawnych sceptyków…

1. Jądrem wielu, znanych od starożytności, rozumowań sceptyckich jest pojęcie nieskończonego regresu, tj. nieskończonego ciągu procedur uzasadniających.

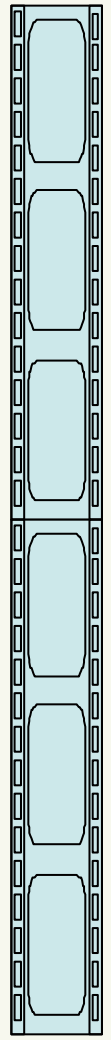
- **Przykładowo:**

  a) nieskończony regres kryteriów prawdziwości

  b) nieskończony regres przesłanek w dedukcji

  c) nieskończona liczba obserwacji w indukcji

  d) nieskończony ciąg podziałów w analizie zjawiska ruchu



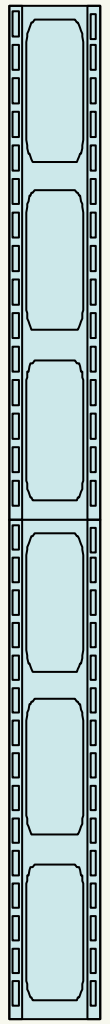
Nieskończoność w informatyce

2. Na polu informatyki (a dokładniej: w odniesieniu do programów komputerowych) nieskończoność można rozumieć na (co najmniej) trzy sposoby:

a) niekończący się czas działania programów (–)
[*poziom działania*]

b) nieskończona liczba modułów oprogramowania (?)
[*poziom struktury*]

c) nieskończona liczba problemów rozwiązywanych za pomocą skończonego programu (+)
[*poziom efektów*]

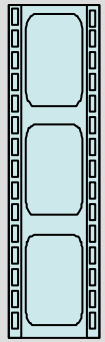


Problemy nieobliczalne

3. Dobrze znanym przejawem kłopotów z nieskończonością typu (2a) są problemy cyfrowo nieobliczalne, czyli:

a) nieobliczalne zasadniczo
(*kanoniczny przykład: problem stopu*)

b) nieobliczalne praktycznie
(*typowy przykład: problem komiwojażera*)

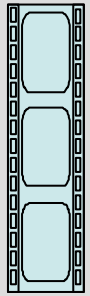


- Istota pierwszych polega na zasadniczej (i matematycznie dowiedzionej) **niemożności** opracowania ogólnego algorytmu, który rozwiązałby dany problem we wszystkich przypadkach szczególnych.
- Istota drugich polega na zbyt dużej **złożoności** czasowej i/lub pamięciowej możliwych do zastosowania algorytmów: gdy rozmiar danych wejściowych rośnie, to liczba niezbędnych operacji i/lub zasobów **rośnie** w sposób nieporównywalnie szybszy (praktycznie zaś: uniemożliwiający rozwiązanie).

Nieobliczalność a nieskończoność

4. Problemy nieobliczalne **zasadniczo** mają charakter infinitystyczny, ponieważ w pewnym sensie są sprowadzalne do kanonicznego problemu stopu;

   ten zaś polega na niemożności algorytmicznego stwierdzenia, czy dany program dla określonych danych początkowych zatrzyma się, czy też będzie działał w nieskończoność.



5. Problemy nieobliczalne **praktycznie** mają charakter infinitystyczny, ponieważ potrzebny na ich rozwiązanie czas jest (dla odpowiednio dużych danych wejściowych) nierozsądnie długi, tj. praktycznie nieskończony.

Zasadniczo czy tylko cyfrowo?

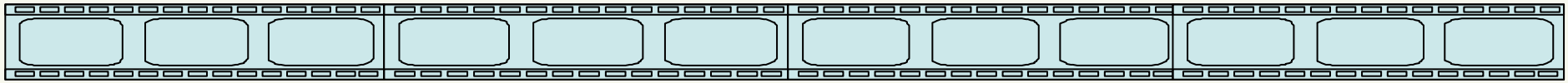
4’. (Sceptyczne) wątpliwości wzbudzane przez fakt istnienia problemów zasadniczo nieobliczalnych dotyczą tylko technik cyfrowych (tj. turingowskiego modelu obliczeń).

► **Tymczasem:**

a) problemy nieobliczalne zawierają podproblemy obliczalne cyfrowo,

b) istnieją inne modele obliczeń pozwalające pokonać barierę cyfrowej nieobliczalności (w tym model analogowy),

c) ludzki umysł może mieć wrodzoną zdolność pokonywania kolejnych barier obliczalności, w tym wynajdywania odpowiednich technik informatycznych.



Praktycznie czy deterministycznie?

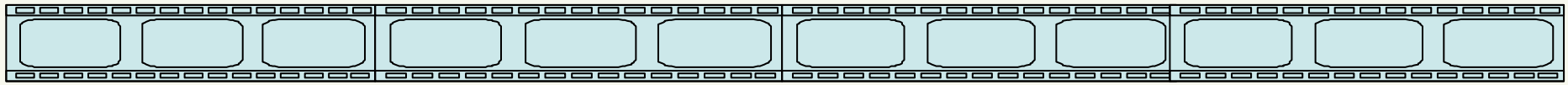
5’. (Sceptyczne) wątpliwości wzbudzane przez fakt istnienia problemów praktycznie nieobliczalnych dotyczą technik cyfrowych, deterministycznych, a także niezawodnych (na poziomie rozwiązania).

- **Tymczasem:**

  a) istnieją modele obliczeń niedeterministycznych (w tym kwantowych)

  b) istnieją różne techniki heurystyczne

  c) istnieją programy uczące się



Programy uczące się

6. Fakt istnienia problemów nieobliczalnych praktycznie, w tym problemów o zbyt dużej złożoności czasowej, stanowi ważną przyczynę badań nad:

a) strategiami heurystycznymi,

b) programami uczącymi się (PU),
tj. zmieniającymi swój kod w interakcji ze środowiskiem.

Wykorzystywane w PU strategie uczenia się można podzielić na:

a) **logicystyczne** – polegające na informatycznej realizacji formalnych schematów wnioskowań,

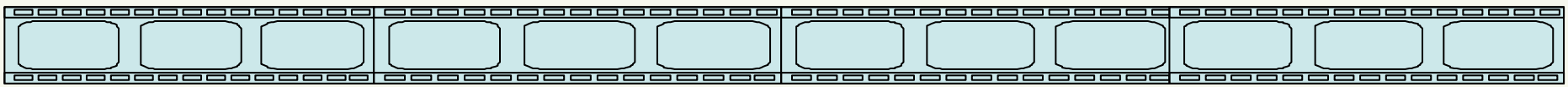
b) **naturalistyczne** – inspirowane obserwacją naturalnych układów do przetwarzania informacji, np. na poziomie mechanizmów biologicznych.

Nieskończony regres PU

7. Teoretyczna analiza struktury i funkcji programów uczących się (PU) prowadzi do poważnego problemu infinitystycznego, który dotyczy struktury PU.

Oto problem:

Za efektywną interakcję PU z otoczeniem odpowiadają korzystne zmiany jego wewnętrznych parametrów, które to zmiany kontroluje **program wyższego rzędu**.
Gdyby program wyższego rzędu miał również podlegać korzystnym zmianom, to musiałyby kierować tym programy jeszcze wyższego rzędu…
To zaś rodzi groźny (i teoretycznie nieusuwalny) problem **nieskończonego regresu** kolejnych programów uczących się.



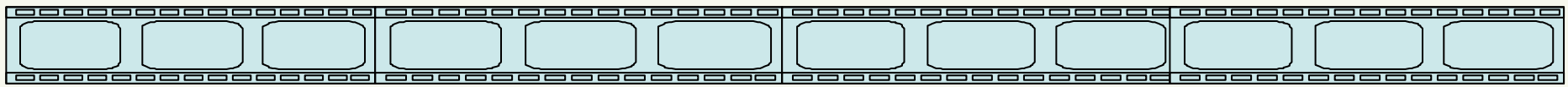
Regres PU w praktyce

7'. W programistycznej praktyce nieskończony regres PU nie może wystąpić;
ciąg schematów uczenia się najczęściej kończy się już na pierwszym poziomie.

Ostatni ze schematów uczenia się może mieć charakter dwojaki:

a) deterministyczny
[*przykład: generalizacja + zapamiętywanie*]

b) niedeterministyczny
[przykład: algorytmy genetyczne]



Nieskończony regres PE

8. Efektywnym schematem uczenia się indeterministycznego jest schemat ewolucyjny (oparty np. na AG).

W jego przypadku ponownie jednak pojawia się problem nieskończonego regresu, dotyczący tym razem funkcji oceny generowanych ewolucyjnie rozwiązań.

Oto problem:

Za efektywność ewolucyjnej strategii uczenia się odpowiada **funkcja celu**, pozwalająca oceniać generowane próbnie rozwiązania (funkcja modeluje własności środowiska, w którym uczy się program).

Gdyby uczenie się jednak miało obejmować cele (a nie tylko metody ich osiągania), to należałoby opracować program ewolucyjny **wyższego rzędu**, generujący optymalne cele na podstawie nowej funkcji celu.

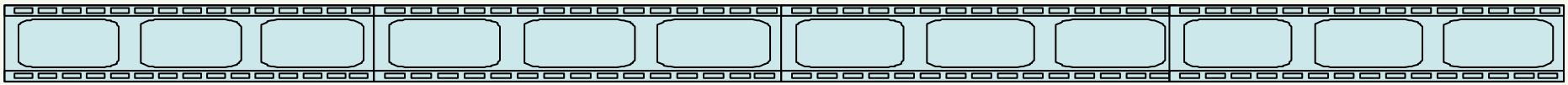
Kolejna funkcja celu musiałaby być generowana za pomocą programu ewolucyjnego jeszcze wyższego rzędu…

Regres PE w praktyce

8’. W programistycznej praktyce nieskończony regres PE (tj. funkcji oceny) nie może wystąpić;

większość PE rozwiązuje konkretne problemy, a wtedy funkcję oceny można utożsamić z opisem problemu.

Gdyby jednak PE miały być maksymalnie wszechstronne, to dla uniknięcia nieskończonego regresu, trzeba by przyjąć istnienie ostatecznego systemu wartości, względem którego PE byłyby oceniane.



Pytania filozoficzne

9. Informatyczne kłopoty z nieskończonością prowadzą zatem do ważkich pytań o naturę procesów i nośników niezbędnych do zrealizowania efektywnych technik informatycznych (ostatecznie zaś: do pytań o naturę rzeczywistości i podstawę ludzkich działań):

a) dyskretność vs ciągłość,
b) determinizm vs indeterminizm
c) podstawowy system wartości

Ich rozstrzygnięcie zdaje się przesądzać o wyborze typu maszyn usprawniających naszą zdolność do rozwiązywania problemów.