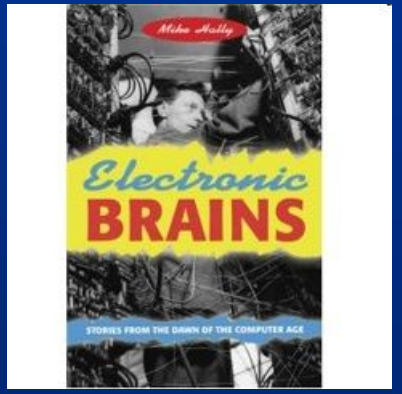
Sztuczna Inteligencja
Programy oparte na szukaniu

Włodzisław Duch
Katedra Informatyki Stosowanej UMK
Google: Wlodzislaw Duch

Co było:

- Szukanie ślepe.
- Szukanie heurystyczne.
- Szukanie niedeterministyczne.

Co będzie



Pierwsze programy AI - mają głównie znaczenie historyczne, więc to tylko rzut oka. Te programy pozwoliły lepiej zrozumieć wyzwania stojące przed AI.

- **Teoretyk logiki**
- **General Problem Solver**
- **Geometra**
-

"It is not my aim to surprise or shock you - but the simplest way I can summarize is to say that there are now in the world machines that can think, that can learn and that can create. Moreover, their ability to do these things is going to increase rapidly until - in a visible future - the range of problems they can handle will be coextensive with the range to which the human mind has been applied."

Simon & Newell 1958

Teoretyk logiki

Logic Theorist (A. Newell, J.C. Shaw, H.A. Simon, 1956)

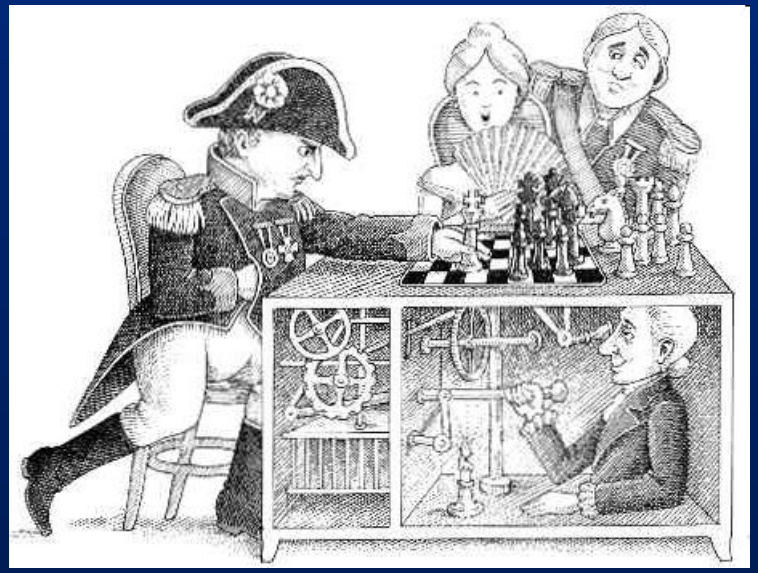
Jeden z pierwszych projektów AI opartych na algorytmach szukania.

Niestety nie można już znaleźć działającego programu (pisany w IPL2).

LT dowodził twierdzeń z *Principia Mathemathica* Whitheada i Russela (1910) dotyczących rachunku zdań.

Opiera się na 5 aksjomatach:

1. $(p \vee p) \Rightarrow p$
2. $p \Rightarrow (q \vee p)$
3. $(p \vee q) \Rightarrow (q \vee p)$
4. $[p \vee (q \vee r)] \Rightarrow [q \vee (p \vee r)]\}$
5. $(p \Rightarrow q) \Rightarrow [(r \vee p)] \Rightarrow (r \vee q)]$



Teoretyk logiki - operatory

3 operatory redukcji:

- Oderwanie: by pokazać X szukaj A $\Rightarrow$ X i dowiedź A.
- Łączenie w przód: by pokazać X postaci A $\Rightarrow$ C poszukaj coś (aksjomat, twierdzenie) postaci A $\Rightarrow$ B i dowiedź B $\Rightarrow$ C.
- Łączenie w tył: by pokazać X postaci A $\Rightarrow$ C poszukaj coś w postaci B $\Rightarrow$ C i dowiedź A $\Rightarrow$ B.
- Podstawienie:

  np. p $\Rightarrow$(q $\vee$ p) podstawiamy p $\vee$ q zamiast p i mamy

  (p $\vee$ q) $\Rightarrow$[q $\vee$ (p $\vee$ q)].

  Stosuj te operatory do wyczerpania pamięci lub problemów.

Zakończ: porównaj ze znanymi twierdzeniami i aksjomatami; badaj ogólne podobieństwo wyrażeń, porządkuj wg. prostoty.

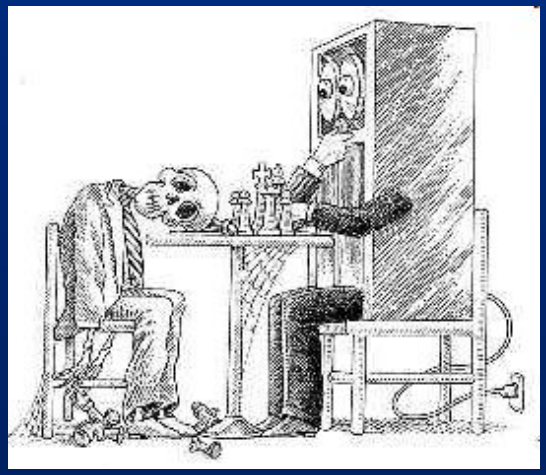
Jeśli to nie koniec dodaj kolejny problem cząstkowy.

Teoretyk logiki - działanie

Wymiana: operator $\Rightarrow$ wymienić można zgodnie z jego definicją w rachunku zdań tj:

$p \Rightarrow q$ zamień na $\neg p \vee q$

Strategia: przeszukujemy przestrzeń stanów na ślepo, rozumując do tyłu.



Typowe twierdzenia dowodzone przez LT:

2.01 $(p \Rightarrow \neg p) \Rightarrow \neg p$

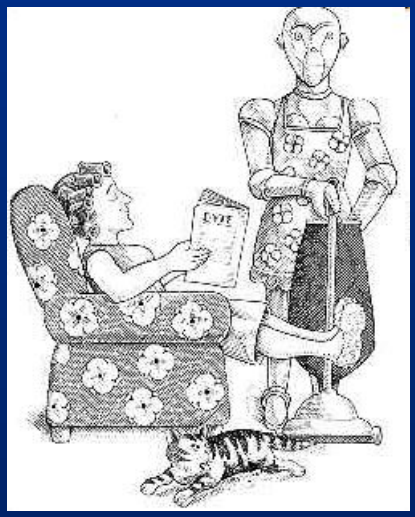
2.31 $[p \vee (q \vee r)] \Rightarrow [(p \vee q) \vee r)]$

2.45 $\neg (p \vee q) \Rightarrow \neg p$

Z 52 twierdzeń z *Principia Mathematica* LT udowodnił 38, na bardzo prymitywnych komputerach; są one stosunkowo łatwe.

Dla twierdzenia 2.85 LT znalazł ciekawszy dowód niż podany w książce PM. Kopia kodu w GitHub w muzeum programów dowodzących twierdzenia.

General Problem Solver (GPS) czyli Ogólny Rozwiązywacz Problemów



Newell, J.C. Shaw, H.A. Simon, rozwijany od 1957 roku.

Dwa cele GPS:

Rozwiązywanie problemów wymagających inteligencji.

Stworzenie teorii rozwiązywania problemów przez ludzi.

GPS składał się z:

- abstrakcyjnego rozwiązywacza problemów;
- wiedzy o zadaniu, zawartej w strukturach danych, tworzących „środowisko problemu”.

Obiekty i dozwolone transformacje – struktury danych GPS.

Zadanie GPS: przekształcić stan początkowy w końcowy.

GPS - cel

- Cel: struktura danych, zawiera obecną sytuację, żądaną sytuację, historię przekształceń wykonanych na obecnej sytuacji by dojść do pożądanej.

- Cele zawierają 3 typy działań:

  $A \rightarrow B$,

  redukcja różnicy A i B przez modyfikację A,

  zastosowanie operatora O(A).

- Zadanie: transformacja z A do B.

Np. $L1 = \{ R \wedge (\neg P \Rightarrow Q)\} \Leftrightarrow L0 = \{(Q \vee P) \wedge R \}$

Celem jest przekształcenie problemu L1 w L0

GPS - heurystyki

Heurystyki:

- Każdy cel powinien być prostszy niż cel wyjściowy.
- Nie należy powtarzać takich samych celów.
- Nowy obiekt (cel) nie powinien być dużo większy niż cel początkowy.

Różne metody szukania:

- analiza ś**rodków-celów** (means-ends analysis);
- typy różnic cel/stan bieżący;
- działanie operatorów w/g typów różnic.

Szukanie w głąb, po jednym operatorze, gdy robi się zbyt trudno (stan bieżący zbytnio się komplikuje) cofa się.

Transformacje obiektów by operator był stosowalny.

Program „ukierunkowany jest na cel”.

Historia szukania zapisywana jest w grafie typu AND/OR.

GPS - działanie

12 operatorów reprezentujących reguły wnioskowania, np:

1. $A \vee B \rightarrow B \vee A$
2. $A \wedge B \rightarrow B \wedge A$
3. $A \vee B \leftrightarrow \neg(\neg A \wedge \neg B)$
4. $A \Rightarrow B \leftrightarrow \neg A \vee B$

Zdefiniowano różnice, od trudnych do łatwych:

1) Występowanie zmiennej tylko w jednym z wyrażeń.
2) Występowanie zmiennej różną liczbę razy.
3) Różnice w znaku.
4) Różnice w użyciu $\wedge$
5) Różnice w położeniu składowych w wyrażeniach.

GPS - rezultaty

GPS: heurystyczne reguły miały być uniwersalne, niezależne od problemu. Reprezentacja obiektów i operatorów nie dała się całkiem uniezależnić od rodzaju problemów.

Początkowo GPS rozwiązał tylko dwa problemy pozalogiczne.

Rozszerzenie Newell, Ernst (1969) zwiększyło możliwości.

Opis problemu przy pomocy list ograniczeń, więzów, ulepszona reprezentacja operatorów i kilka innych usprawnień.

W nowej wersji program rozwiązał zadania z 11 dziedzin, np. gier, całkowania symbolicznego, dowodzenia twierdzeń; niestety działał gorzej w problemach logicznych.

Ilustracja działania GPS, kod w python na Githubie.

GPS - przykład

Analiza celów i środków: szukaj dostępnej metody.

Jak dojechać do lotniska?

Jaka jest różnica pomiędzy stanem obecnym a pożądanym?

250 km.

Jakie są środki by zmniejszyć taką różnicę?

Autobus, pociąg, samochód.

Jak znaleźć rozkład autobusów?

Pod http://www.pks.pl ... może nie w tamtych czasach.

Czy połączenie jest zadawalające?

Znaczenie GPS polegało na zrozumieniu trudności AI, program ewoluował, zastąpiony przez architekturę SOAR, rozwijaną do teraz.

Było to pierwsze przybliżenie do zrozumienia ludzkiego myślenia, wprowadzające heurystyki jako podstawę rozumowania.

Geometra

Herbert Gelernter, 1959, IBM New York.

Zadanie: udowadnianie twierdzeń geometrycznych na poziomie szkoły średniej; program napisany w Fortranie;
pierwszy program z koniunkcją podcelów.

Aksjomaty = operatory redukujące problem.
Np. przystawanie $\Delta \Leftrightarrow$ jedna strona i 2 kąty lub

2 boki i jeden kąt są jednakowe.

Rozumuje wstecz, drzewo podproblemów AND/OR.

Startuje z opisu problemu i zbioru współrzędnych punktów.

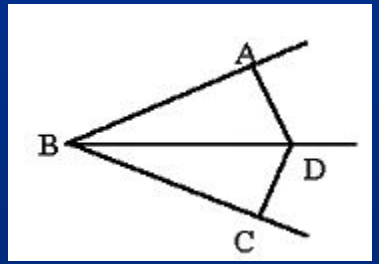
Dowodem jest zbiór stwierdzeń redukujących cel do trywialnego lub aksjomatu. Redukcja grafu szukania wykonywana jest za pomocą reprezentacji geometrycznej problemu – nie każda transformacja jest możliwa.

Geometra: przykład problemów

Dane: Kąt ABD=DBC

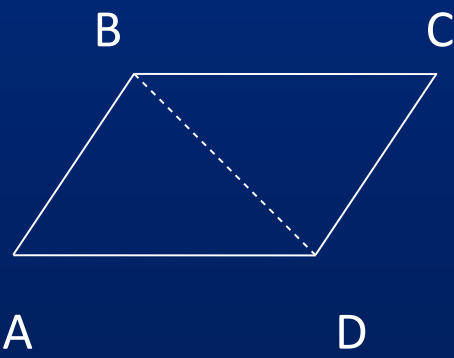
AD prostopadłe do AB

DC prostopadłe do BC



ABCD jest rombem, odcinek BC jest równoległy do AD a odcinek BC jest równy AD.

Pokaż, że odcinek AB równy jest CD.



Tworząc cele należy odrzucać wcześniejsze cele by nie popaść w rozumowanie kołowe.

Warto rozpoznawać symetrię i posługiwać się diagramem.

Inne wczesne projekty

SAINT=Symbolic Automatic INTegrator
J. Slagle, 1961, praca doktorska, MIT.

Całkowanie symboliczne, przekształcanie wyrażeń.
Program napisany w LISPie, konieczne są heurystyki.

Reprezentacja redukcji problemów do podproblemów.

Rozwiązał 84 z 86 zdań z egzaminu na MIT.

Np: całka z $(\sec^2 t)/(1+\sec^2 t - 3\tan t)\ dt$

Po usprawnieniu rozwiązał też i $\cos(x^{1/2})dx$, $x(1+x)^{1/2}\,dx$

SIN, Symbolic Integration, 1967, J. Moses, MIT.

SIN rozwiązywał najtrudniejsze całki.

Powstała z tego MACSYMA a potem Mathematica.

Inne projekty cd

STRIPS (R. Fikes, N. Nilsson, 1971, SRI International)

Planowanie ruchów robota w pokoju ze skrzynkami i pudłami.

Model świata, plan ruchów.

Opis stanu: za pomocą rachunku predykatów.

Operatory: akcje robota.

Sprawdzanie wstępnych warunków stosowalności.

Wynikiem działania jest zmiana modelu świata.

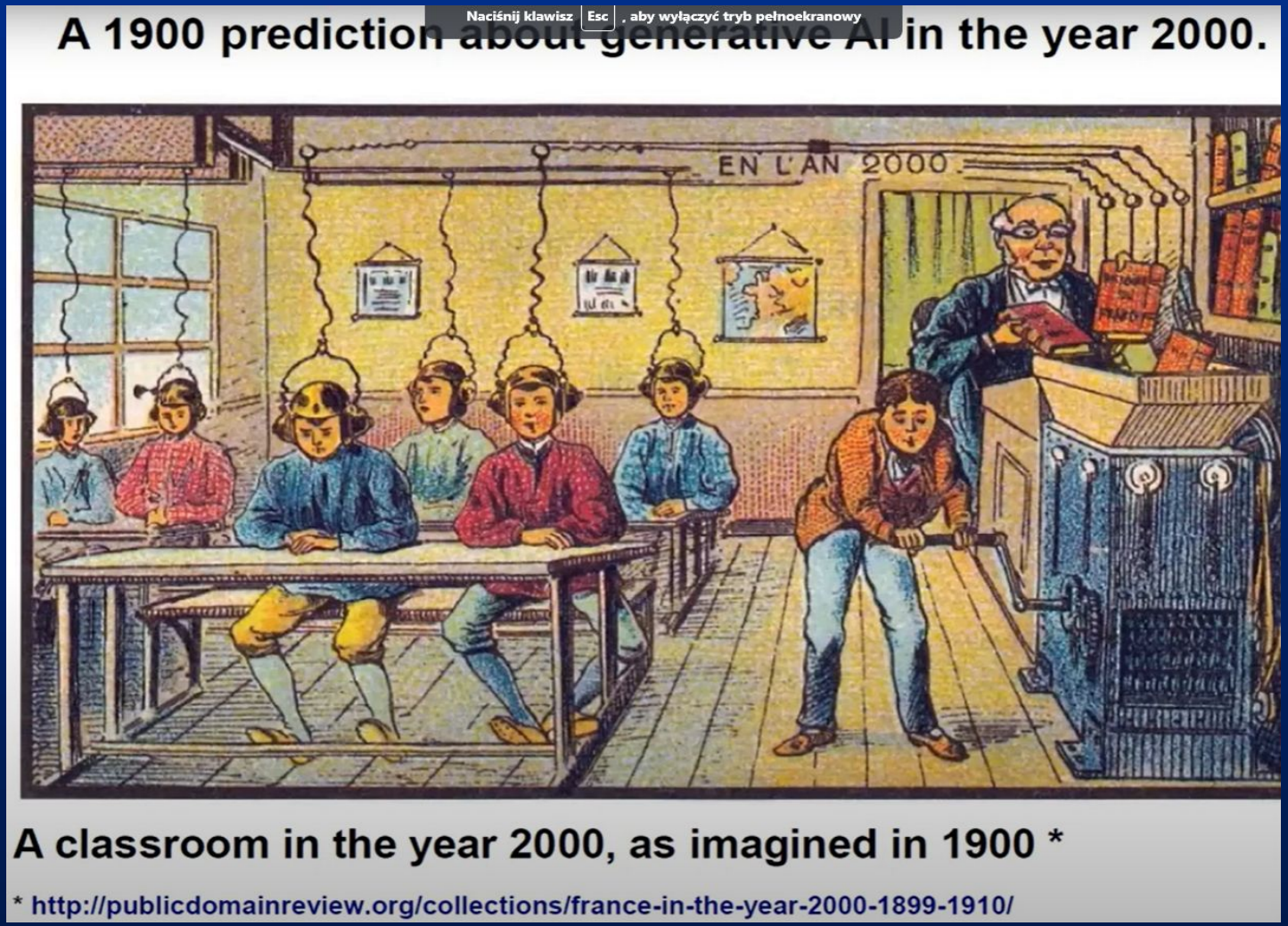
ABSTRIPS (E. Sacerdoti, 1974)

Ulepszenia: hierarchiczne planowanie, czyli najpierw szkic działania, a potem tworzy się plan szczegółowy.

Wydawało się, że skalowanie wszystko załatwi …

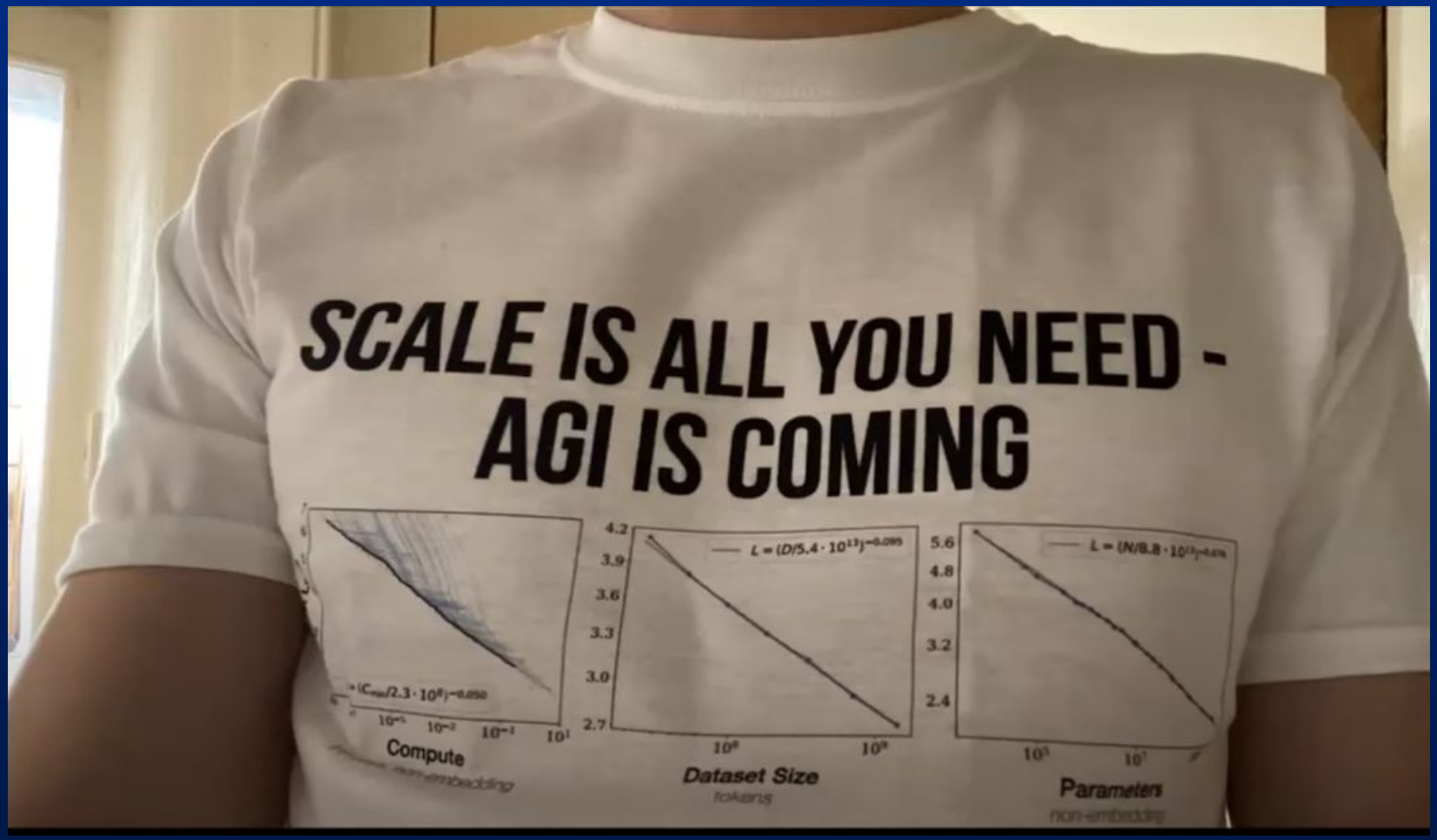
Skalowanie

Większe bazy, szybsze maszyny => inteligencja?



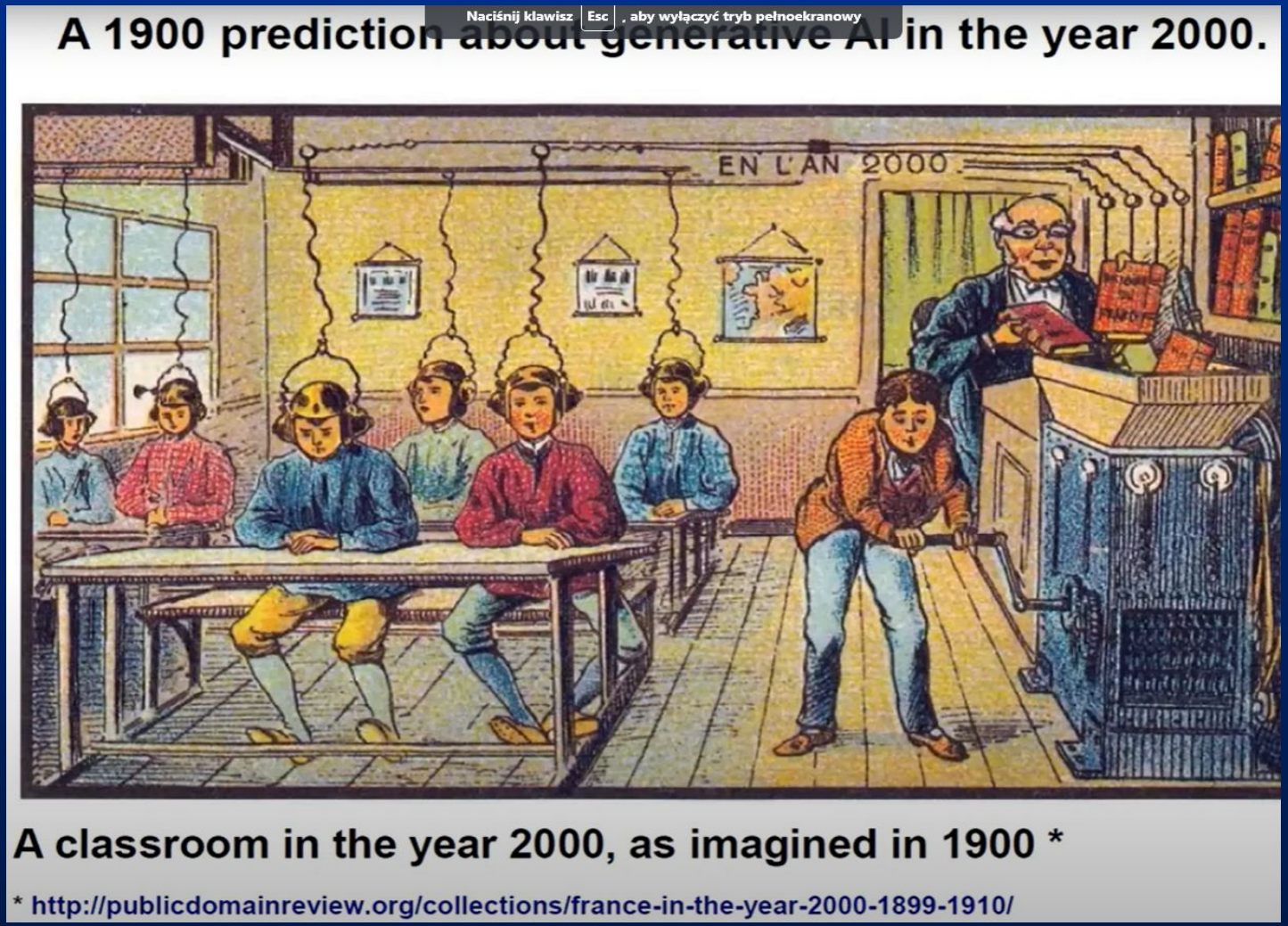
Skalowanie to wszystko?

M Harmel, A Paras, A Pasternak, N Roy, G Linscott, Scaling Is All You Need: Autonomous Driving with JAX-Accelerated Reinforcement Learning arXiv 12/2023.



Skalowanie

Większe bazy, szybsze maszyny => inteligencja?



Skalowanie

Większe bazy, szybsze maszyny => inteligencja?