

# Sztuczna Inteligencja

Soma Dutta

Wydział Matematyki i Informatyki, UWM w Olsztynie  
[soma.dutta@matman.uwm.edu.pl](mailto:soma.dutta@matman.uwm.edu.pl)

Wykład - 1: Wprowadzenie

Semestr letni 2020

# Sztuczna inteligencja, początkowe podejście Alana Turinga

Alan Turing (1950): Computing machinery and intelligence

- ▶ Czy maszyny mogą myśleć? Czy maszyny mogą zachowywać się inteligentnie?
- ▶ **Turing test** ([https://mfiles.pl/pl/index.php/Test\\_Turinga](https://mfiles.pl/pl/index.php/Test_Turinga)): Turing zdefiniował, że maszyna jest inteligentna o ile jest w stanie przejść następujący test:
  - ▶ Osoba A siedzi w zamkniętym pokoju z dwoma terminalami komputerowymi.
  - ▶ Jeden terminal jest podłączony do maszyny, a drugi do osoby B. Osoba A nie wie, który jej terminal jest podłączony do maszyny, a który do osoby B.
  - ▶ Osoba A może zadawać pytania na obu terminalach. Zadanie osoby A polega na podjęciu decyzji, który terminal należy do maszyny.
- ▶ Maszyna jest inteligentna jeśli osoba A nie jest w stanie odróżnić (w określonym czasie) maszyny od osoby B.

- ▶ Obecnie istnieje wiele chatterbotów pracujących online, które zwykle nie są w stanie ukryć swojej sztuczności co okazuje się już po krótkim czasie.
- ▶ (Zobacz: <https://www.spidersweb.pl/2018/06/atom2vec-sztuczna-inteligencja.html>)

# Co to oznacza sztuczna inteligencja

Obejmuje różne aspekty: wiedza, wnioskowanie, język, rozumienie, uczenie - to główne wśród nich.

- ▶ Zrozumienie różnych aspektów inteligencji
- ▶ Próby przedstawiania tego w jakiś matematyczny sposób
- ▶ Budowanie systemów, np. modelowanych w postaci agentów, które mogą zachowywać się inteligentnie jak ludzie.

# Z raportu Unii Europejskiej



Brussels, 19.2.2020  
COM(2020) 65 final

## **WHITE PAPER**

### **On Artificial Intelligence - A European approach to excellence and trust**

- ▶ Sztuczna inteligencja odnosi się do systemów, które wykazują inteligentne zachowanie poprzez analizowanie ich środowiska i podejmowanie akcji, do pewnego stopnia w sposób autonomiczny, w celu uzyskania specyficznych celów.

## Z jednego artykułu napisanego przez znanych autorów

- ▶ **The Mathematics of Learning: Dealing with Data: Tomaso Poggio and Steve Smale**
- ▶ Systemy sztucznej inteligencji to systemy oprogramowania (mogą to być również systemy sprzętowe) zaprojektowane przez ludzi, które mając do zrealizowania złożony cel, działają w wymiarze fizycznym i cyfrowym przez postrzeganie swego środowiska, poprzez pozyskiwanie danych, interpretowanie strukturalnych i niestrukturalnych danych, wnioskowanie o wiedzy lub przetwarzanie informacji wywodzonych z danych i decydowanie o wyborze najlepszych akcji do wykonania w celu osiągnięcia celu. Systemy sztucznej inteligencji mogą używać reguł symbolicznych lub wyuczać się modeli numerycznych, mogą również adaptować swe zachowanie przez analizę wyniku ich poprzedniego oddziaływania na środowisko.

# Trochę historii: Rozwój różnych aspektów sztucznej inteligencji

Okres	Rozwój
1930-1950	(1930) Kurt Gödel: Udowodniono pewne twierdzenia dotyczące wnioskowania logicznego, znane jako 'twierdzenie o kompletności logiki pierwszego rzędu',
	(1936) Alan Turing: 'problemu stopu programu'
	(1950) Newel, Simon: (Na podstawie twierdzenia Goedle) pierwszy automatyczny system dowodzenia twierdzeń
	(1940) McCulloh, Pitts: Przedstawiono pierwszy model matematyczny dla sieci neuronowych
	(1950) Artykuł Turinga: Computing Machinery and Intelligence
1956	Spotkanie w Dartmouth: powstaje termin 'Sztuczna Inteligencja'
	McCarthy: Wprowadzono LISP, język, który może przetwarzać struktury symboliczne
1960 - 1990	(1965) Lotfi A. Zadeh: Zbiór rozmyte
	(1970) PROLOG, Język programowania został wprowadzony
	Wykazano, że sieci neuronowe są w stanie uczyć się na przykładach treningowych
	Nettalk, system, który był w stanie nauczyć się mowy z przykładowych tekstów (w oparciu o rozpoznawanie wzorców)
	System hybrydowy łączący sieć neuronową i logikę rozmytą
(1982) Teoria zbiorów przybliżonych, Z. Pawlak	
1990 - dotychczas	Eksploracja danych, Data science - ma na celu pozyskiwanie informacji z dużych zbiorów danych i bazy-wiedzy
	Rozproszone autonomiczne agenty, których celem jest rozwiązywanie problemów we współpracy wielu agentów

# Co rozumiemy przez agenta

- ▶ Agent to obiekt, który może **postrzegać swoje środowisko** za pomocą **sensorów** i **oddziaływać** na to środowisko za pomocą **aktuatorów**.
- ▶ Na przykład:
  - ▶ Agent ludzki (Human agent) ma **oczy, uszy i inne narządy zmysłów jako sensory** i **dłonie, nogi, drogi głosowe jako aktulatory**.
  - ▶ Agent robota (Robotic agent) ma **kamerę, dalmierze na podczerwień jako sensory** i **różne silniki jako aktulatory**.
  - ▶ Agent oprogramowania (Software agent) za pomocą **naciśnięć klawiszy, zawartości plików, pakietów sieciowych realizuje zadania sensorów** i **przez wyświetlanie na ekranie, tworzenie zapisów plików, wysyłanie pakietów sieciowych realizuje zadania aktuatorów**.



# Postrzeganie środowiska przez agenta

Duża dziedzina: Nauki kognitywne (Cognitive Science)

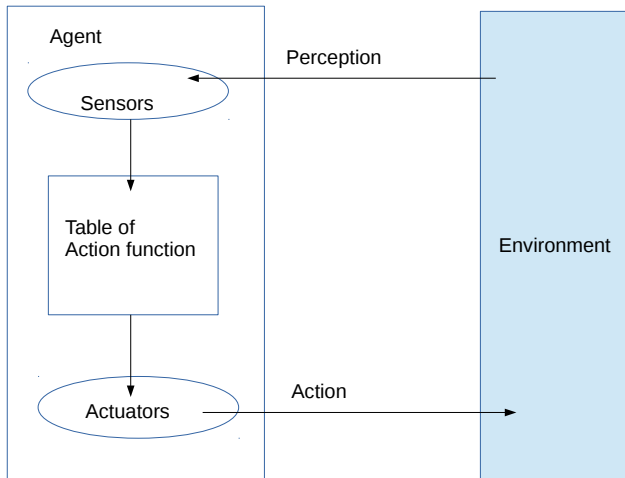
- ▶ **Percept**: Odnosi się to do danych percepcyjnych agenta w danym momencie. Jest to wejście, które agent postrzega w danym momencie.
- ▶ **Sekwencja percepcji**: Oznacza pełną historię postrzeganą przez agenta.

# Działanie agenta

- ▶ Wybór działania przez agenta zależy od całej dotychczas obserwowanej sekwencji percepcji.
- ▶ Matematycznie opisuje to funkcja agenta, która przypisuje akcję do zadanej sekwencji percepcji.
- ▶ Funkcję tę można również przedstawić jako tabelę ze wszystkimi możliwymi sekwencjami percepcji (o skończonej długości) i odpowiednimi działaniami podejmowanymi przez agenta.

# Pierwszy krok do zaprojektowania inteligentnego agenta

- ▶ **Task environment** (Środowisko zadań): Opis środowiska zadań według **PEAS** (**P**erformance (Wydajność), **E**nvironment (Środowisko), **A**ctuators (Aktuatory), **S**ensors (Sensory)).

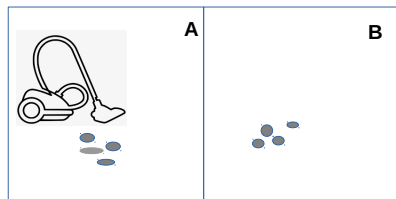


## Przykład: inteligentny agent

Świat odkurzacza (Vacuum cleaner world):

- ▶ Załóżmy, że ma dwie lokalizacje: kwadrat A i kwadrat B.
- ▶ Odkurzacznik rozpoznaje, w którym kwadracie znajduje się i czy na tym miejscu jest brud.
- ▶ Może wybrać ruch w lewo, ruch w prawo, wssać brud lub nie robić nic.
- ▶ Prosta funkcja agenta: jeśli bieżący kwadrat jest brudny, to wssać; w przeciwnym razie przejdź na drugi kwadrat.

# Odkurzacz można modelować jak poniżej



Sekwencja percepcji	Akcja
[A, Czysty]	W prawo
[A, Brudny]	Ssuć
[B, Czysty]	W Lewo
[B, Brudny]	Ssuć
[A, Czysty] [A, Czysty]	W prawo
[A, Czysty] [A, Brudny]	Ssuć
⋮	⋮

# Czego więcej potrzebujemy do modelowania inteligentnego agenta

- ▶ **Racjonalność agenta:** Dla każdej możliwej sekwencji percepcji racjonalny agent powinien wybrać działanie, które maksymalizuje miarę jego wydajności, biorąc pod uwagę argumenty dostarczone przez sekwencję percepcji i bazę wiedzy agenta.
- ▶ Więc racjonalność zależy od:
  - (i) pomiaru wydajności
  - (ii) wcześniejszej wiedzy agenta na temat środowiska
  - (iii) działania, które agent może wykonać
  - (iv) dotychczasowej sekwencja percepcji agenta

# Różne rodzaje inteligentnych agentów

- ▶ Task environment: Podział z uwagi na działanie na środowisko zadań
  - ▶ **W pełni obserwowalny** (Fully observable): Jeśli sensory agenta dają pełny stan środowiska
  - ▶ **Częściowo obserwowalne** (Partially observable): Jeśli brakuje niektórych warunków środowiska z powodu szumu lub niedokładności sensora
  - ▶ **Nieobserwowalny** (Unobservable): Agent może nie mieć sensora
- ▶ Podział z uwagi na działanie na podstawie wiedzy o kolejnym stanie środowiska:
  - ▶ **Deterministyczny** (Deterministic): Jeśli następny stan środowiska jest całkowicie określony przez aktualny stan i można zaplanować działanie agenta
  - ▶ **Niedeterministyczny** (Non-deterministic): Gdy niepewność środowiska można opisać jedynie możliwymi wynikami
  - ▶ **Stochastyczny** (Stochastic): Gdy opisana jest niepewność środowiska poprzez przypisanie prawdopodobieństwa do możliwych wyników

# Różne rodzaje inteligentnych agentów

- ▶ Single agent vs. multiagent: **Klasyfikacja na podstawie działania innych agentów**
  - ▶ **Model konkurencji** (Competitive): Gdy jeden agent próbuje zmaksymalizować swoją miarę wydajności w oparciu o zachowanie innego agenta
  - ▶ **Model współpracy** (Cooperative): Gdy współpraca między agentami poprawia wydajność każdego z nich
- ▶ **Klasyfikacja agentów na podstawie warunków akcji**
  - ▶ **Epizodyczny** (Episodic): Gdy doświadczenie agenta jest podzielone na epizody atomowe, a akcja dla każdego odcinka zależy tylko od odpowiedniej sekwencji percepcji w tym odcinku
  - ▶ **Sekwencyjny** (Sequential): Gdy działanie w bieżącym odcinku wpływa na działanie w następnym odcinku
  - ▶ **Stacyczny vs. dynamiczny** (Static vs. dynamic): Jeśli środowisko może się zmienić, gdy agent rozważa/wykonuje akcję, to agent jest dynamiczny; w przeciwnym razie jest statyczny



# Przykłady środowisk zadań

Task environment	Observable	Deterministic	Agent	Episodic	Static
Crossword puzzle (Krzyżówka)	Fully	Deterministic	Single	Sequential	Static
Chess with a clock (Szachy)	Fully	Deterministic	Multi	Sequential	Semi
Poker	Partially	Deterministic	Multi	Sequential	Static
Backgammon (Trik-trak)	Fully	Deterministic	Multi	Sequential	Static
Taxi driving (Jazda taksówką)	Partially	Stochastic	Multi	Sequential	Dynamic
Medical diagnosis (Diagnoza medyczna)	Partially	Stochastic	Single	Sequential	Dynamic
Image analysis (Analiza obrazu)	Fully	Deterministic	Single	Episodic	Semi
Part-picking robot (Robot do pobierania części)	Partially	Stochastic	Single	Episodic	Dynamic
Refinery controller (Kontroler rafinerii)	Partially	Stochastic	Single	Sequential	Dynamic
Interactive English tutor (Interaktywny nauczyciel języka angielskiego)	Partially	Stochastic	Multi	Sequential	Dynamic

# Funkcja agenta kontra program agenta

- ▶ Funkcja agenta jest jak reguła przedstawiona w tabeli i Program agenta implementuje tę funkcję agenta
- ▶ Program agenta jest więc związany z fazą budowy inteligentnego agenta
- ▶ Program agenta możemy przedstawić w następujący sposób

```
function TABLE-DRIVEN-AGENT(percept) returns an action  
persistent: percepts, a sequence, initially empty  
           table, a table of actions, indexed by percept sequences, initially fully specified  
append percept to the end of percepts  
action → LOOKUP(percept, table)  
return action
```

# Składniki programu agenta

Składniki są takie, że znajdziemy odpowiedzi na następujące pytania:

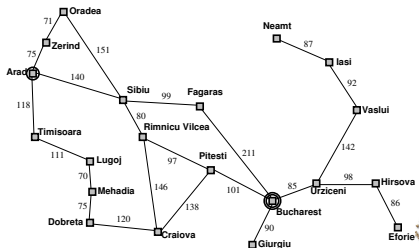
- ▶ Jak teraz wygląda świat? (Obecne środowisko)
- ▶ Jakie działanie należy wybrać? (Wybór akcji)
- ▶ Co będzie wynikiem działania? (wpływ działania)

W oparciu o te komponenty program agenta może być trzech typów:

- (i) Reprezentacja atomowa (Atomic representation)
- (ii) Reprezentacja rozposzona (Factored representation)
- (iii) Reprezentacja ustrukturyzowana (Structured representation)

# Reprezentacja atomowa

- ▶ Każdy stan świata jest niepodzielny i nie ma wewnętrznej struktury
- ▶ Na przykład, znalezienie trasy dojazdu z jednego miejsca do drugiego
- ▶ W szczególności potrzebujemy
  - ▶ algorytmów wyszukiwania (Searching algorithms),
  - ▶ strategii gier (Game strategies)



### Sformułowanie problemu:

- stan początkowy: początkowy stan przed rozwiązaniem problemu
- cel: stan docelowy lub formuła oceniająca, czy dany stan spełnia cel
- rozwiązanie: ciąg akcji prowadzący od stanu początkowego do celu
- koszt rozwiązania: funkcja oceny kosztu rozwiązania równa sumie kosztów poszczególnych akcji występujących w rozwiązaniu

Rozwiązania o niższym koszcie są lepsze niż rozwiązania o wyższym koszcie.

# Reprezentacja rozposzona

- ▶ W tym kontekście, poza samą lokalizacją miasta, skupimy się również na czynnikach, takich jak koszt dotarcia tam - pod względem pieniędzy, odległości, czasu.
- ▶ Tak więc każdy stan zawiera zestaw zmiennych lub atrybutów, z których każdy może mieć określoną wartość
- ▶ Tutaj warto wymienić i poznać:
  - ▶ **Algorytmy spełniające więzy** (Constrained satisfaction algorithms)
  - ▶ **Rachunek zdań** (Propositional logic)
  - ▶ **Planowanie** (Planning)
  - ▶ **Sieci bayesowskie** (Bayesian network)
  - ▶ **Algorytmy uczenia maszynowego** (Machine learning algorithms)

# Reprezentacja ustrukturyzowana

- ▶ W tym kontekście mamy do czynienia z sytuacjami, w których oprócz indywidualnych informacji o różnych stanach, mamy różne relacje między stanami
- ▶ Tutaj warto wymienić i poznać:
  - ▶ **Rachunek pierwszego rzędu** (First order logic)
  - ▶ **Uczenie się oparte na wiedzy** (Knowledge-based learning)
  - ▶ **Przetwarzanie języka naturalnego** (Processing of natural language)

Łącząc różne aspekty, skoncentrujemy się na następujących tematach

1. Algorytmy wyszukiwania (w tym heurystyczne)
2. Strategie w grach
3. Wnioskowanie w logice
4. Sieci neuronowe
5. Zbiory rozmyte
6. Systemy decyzyjne i uczące się: Zbiory przybliżone, reguły decyzyjne
7. Sieci bayesowskie



Dziękuję za uwagę