

# Sztuczna Inteligencja

## Reprezentacja wiedzy V

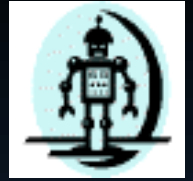
### Agenci

Włodzisław Duch

Katedra Informatyki Stosowanej UMK

Google: Włodzisław Duch

# 3 prawa robotyki



3 prawa robotyki Issaca Asimova  
(z Podręcznika Robotyki, wyd. 56, AD 2058)

- Robot nie może skrzywdzić człowieka swoim działaniem, jak też nie może go skrzywdzić przez zaniechanie działania, które mógłby podjąć.
- Robot musi zawsze wykonywać polecenia człowieka, poza przypadkami, w których zabrania mu tego Pierwsze Prawo.
- Robot musi chronić siebie przed uszkodzeniami we wszystkich przypadkach, w których nie zabrania mu tego Pierwsze lub Drugie prawo.

Taki robot musi mieć naprawdę głębokie zrozumienie świata ...

# Agenci



Agent programowy to:

- Każdy system, który odbiera informacje z otoczenia i reaguje na te informacje.
- Cel: stworzyć agentów, którzy są samowystarczalni, zdolni do wykonywania wyspecjalizowanych funkcji.

Człowiek:

Sensory: oczy, uszy,  
skóra, nos, język ...

Efektory: ręce, nogi,  
język, usta ...

Robot:

Sensory: kamery, czujniki  
podczerwieni, sonary,  
mikrofony ...

Efektory: koła, motory,  
manipulatory ...

Programy:

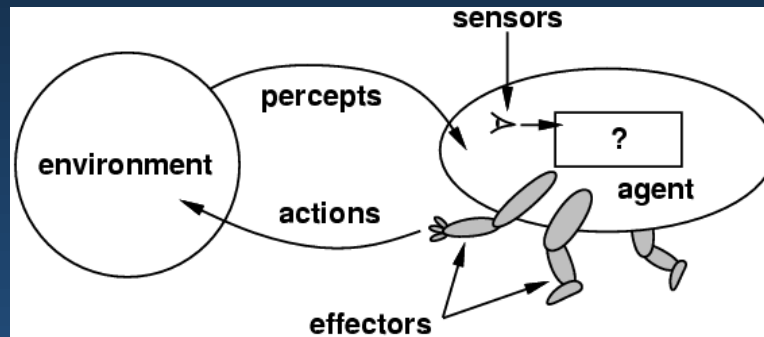
Sensory wirtualne:  
ciągi bitów, zawierające  
informacje dochodzące  
ze środowiska ...

Efektory wirtualne:  
ciągi bitów, przesyłane do  
środowiska...

# Działanie agenta

Idealny racjonalny agent powinien:

- Posiadać miarę oceny swojego działania z punktu widzenia stawianych przed nim celów.
- Wykorzystywać informację zawartą w nadchodzących postrzeganych danych tak, by optymalizować tą miarę w oparciu o dostępną wiedzę.



Idea: Marvin Minsky, Society of mind. Simon & Schuster 1986

Umysł = to co robi mózg.

Inteligencja: wiele prostych procesów, które ze sobą współpracują.

# Jak?

Działanie racjonalne – oparte na rozumowaniu.

Mając różne możliwości działania, należy wybrać (wyszukać) najlepszą z punktu widzenia celu.

Obiektywne miary jakości działania biorą pod uwagę.

- Szansa na sukces?
- Czas potrzebny do rozwiązania problemu?
- Koszt tego rozwiązania?

Automatyczny kierowca:

Osiągnięcie celu podróży, przestrzeganie przepisów i bezpieczeństwo jazdy, wybieranie najkrótszej drogi.

Agent wyszukiujący sklepy i porównujący ceny:

Czy znalazł wszystkie oferty? Jak szybko? Czy uwzględnił wszystkie preferencje użytkownika?

# Wiedza i działanie

Działanie racjonalne wymaga wiedzy:

- Wiedza agenta jest zawsze ograniczona, np. z powodu ograniczeń percepcji lub braku doświadczenia, co czasami nie pozwala na znalezienie najprostszego rozwiązania.
- Zachowanie zależy od relacji pomiędzy postrzeganym (senso) a wykonywanym (motorycznych).
- Relacje w prostych przypadkach mogą być zapisane w tabeli.
- Relacje mogą być dane w postaci funkcji.

Agent powinien działać autonomicznie:

- Autonomia wymaga zmiany zachowania w zależności od napływających danych.
- Nabywanie doświadczenia wymaga uczenia się agenta.
- W najprostszym razie uczenie się jest zmianą programu przez twórcę agenta.

# Struktura

Agent = architektura systemu + program

- Program: zbiór algorytmów określających relacje pomiędzy spostrzeżeniami a działaniami agenta.
- Architektura:
  - Sprzęt i środowisko programowe.
  - Dostęp do danych zewnętrznych (percepcji).
  - Kontrola wykonywania programu.
  - Dostęp do efektorów.

Agent scharakteryzowany jest przez:

- Postrzeżenia.
- Akcje.
- Cele.
- Środowisko, w którym działa.

# Przykłady agentów

Agent	Postrzeżenia	Akcje	Cele	Środowisko
Robot przemysłowy	Piksele z kamery	Ruch manipulatora	Sortowanie części	Taśma montażowa
Kontroler rafinerii	Ciśnienie, temp, par. chemiczne	Regulacja zaworów	Maks. wydajności	Rafineria
Analizator obrazów	Piksele z kamer	Segmentacja obrazu	Prawidłowa segmentacja	Satelita, skrzyżowanie
Diagnostyka medyczna	Testy medyczne	Pytania, testy, terapia	Poprawa zdrowia	Szpital
Automatyczny kierowca	Czujniki, kamery	Kierowanie, wybór drogi	Bezpieczny przejazd	Ulica, ruch drogowy
Agent sklepowy	Ceny, oferty, sklepy.	Negocjacje, wyszukiwanie	Prezentacja, porównanie	Internet



# RoboCup i wielu agentów

Roboty grają w piłkę nożną.

Postrzeżenia ?

Akcje ?

Cele ?

Środowisko ?

Drużyna: system  
wieloagentowy.

Wiele zastosowań.

Cel RoboCup: do 2050 r  
humanoidalne roboty mają  
wygrać z mistrzami świata.

Artemis zwycięzca Robocup



STANDARD PLATFORM LEAGUE



MIDDLE SIZE LEAGUE



RESCUE ROBOT LEAGUE



HUMANOID LEAGUE

# Typy agentów

Agenci mogą być prostymi lub złożonymi systemami.

- Agent wykorzystujący proste odruchy, nie ma pamięci, wykorzystuje reguły definiujące działanie, zwykle korzysta z systemu produkcyjnego (regułowego).
- Agent oparty na tabeli spostrzeżeń/działań, wykorzystuje duże tabele by wybrać następne działanie w zależności od sytuacji; tabela służy tu jako pamięć i umożliwia modelowanie przejść pomiędzy stanami wewnętrznymi.
- Agent oparty na odruchach z pamięcią stanu wewnętrznego, np. agent Markowa pamiętający swój poprzedni stan; tylko stan bieżący ma wpływ na podjętą decyzję.
- Agenci posiadający cele wykorzystują informację o bieżącym stanie i posiadający reprezentację możliwych stanów, ocenianych z punktu widzenia przyjętych celów.
- Agenci maksymalizujący funkcje użyteczności.

# Agent Tablicowy

Agenci posługujący się tablicami korelacji:

**function** TABLE-DRIVEN-AGENT(*percept*) **returns** *action*

**static:** *percepts*, a sequence, initially empty

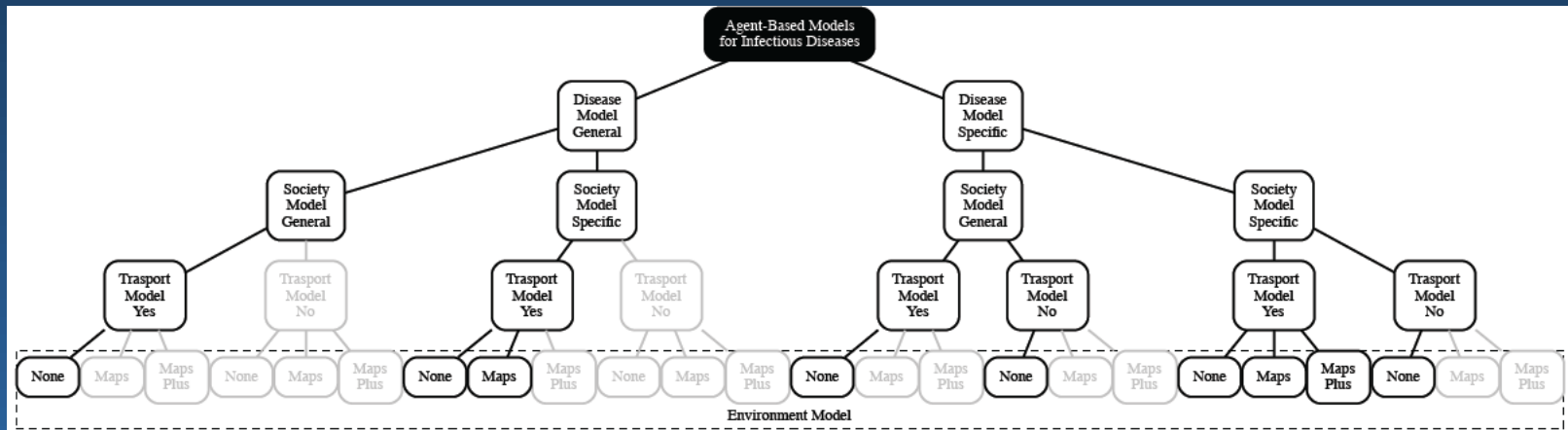
*table*, a table, indexed by percept sequences, initially fully specified

append *percept* to the end of *percepts*

*action* ← LOOKUP(*percepts*, *table*)

**return** *action*

- Agent grający w szachy musiałby mieć tablicę o rozmiarze rzędu  $35^{120}$ .
- Trudno jest zbudować tabelę wszystkich możliwych powiązań.
- Agent nie jest autonomiczny, bo przy zmianie środowiska (np. reguł gry) nie potrafi sobie poradzić, wszystkie akcje trzeba z góry przewidzieć.
- Liczne modele epidemiologiczne wykorzystują technologię wieloagentową.



# Agent z prostymi refleksami

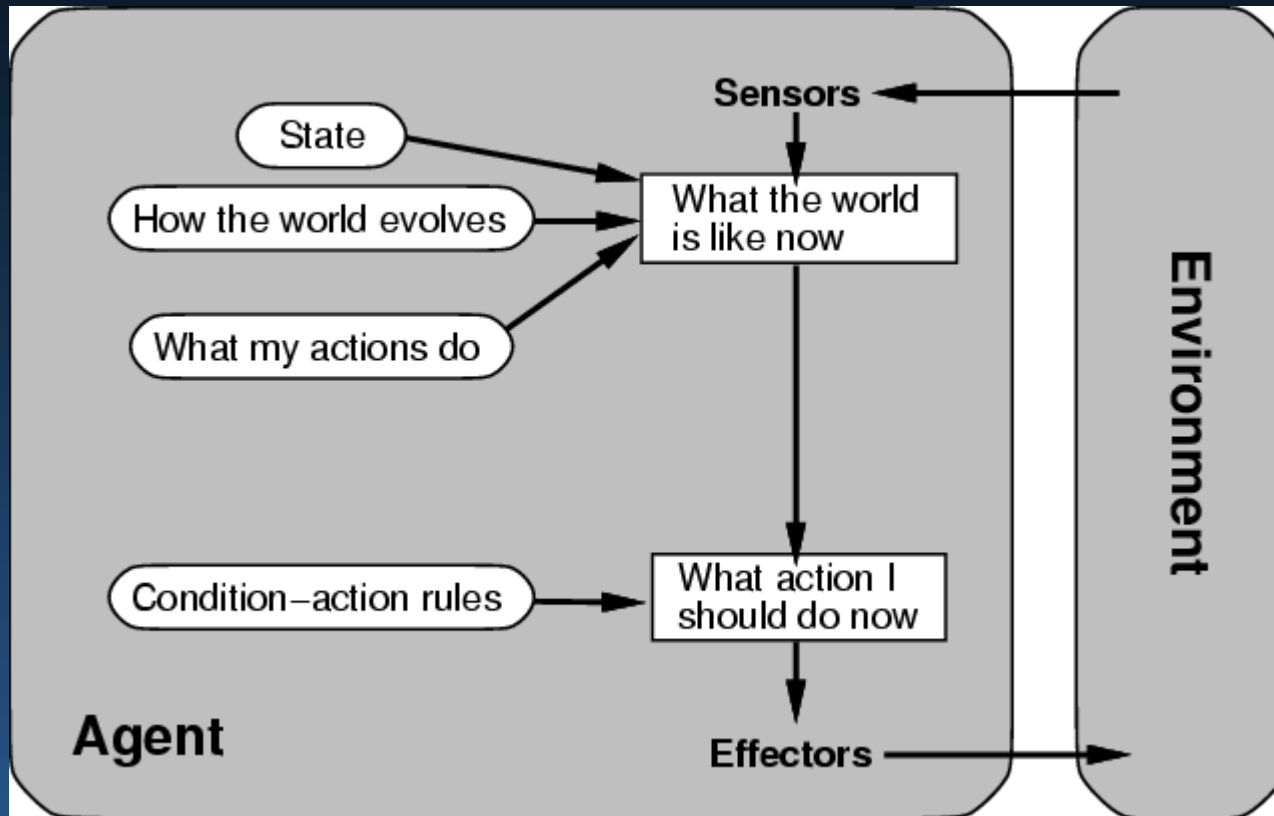
Agenci określający stan i stosujący regułę dla tego stanu:

```
function SIMPLE-REFLEX-AGENT(percept) returns action  
  static: rules, a set of condition-action rules  
  
  state ← INTERPRET-INPUT(percept)  
  rule ← RULE-MATCH(state, rules)  
  action ← RULE-ACTION[rule]  
  return action
```

- Reguły typu: warunek-akcja, np.
  - Jeśli światło(czerwone) to przyspieszaj
  - Jeśli światło(zielone) to hamuj
- Decyzje podejmowane tylko w oparciu o spostrzeżenie, rozważana jest tylko jedna decyzja.
- Zmiana pasa w czasie jazdy wymaga pamięci stanu poprzedniego: czy widać coś we wstecznym lusterku.
- Pamięć jest czasem niezbędna.

# Agent ze stanami wewnętrznymi

Agent ma model środowiska i rezultatów swojego działania.



# Agent ze stanami wewnętrznymi



Agenci zmieniający swoje stany wewnętrzne na skutek percepcji.

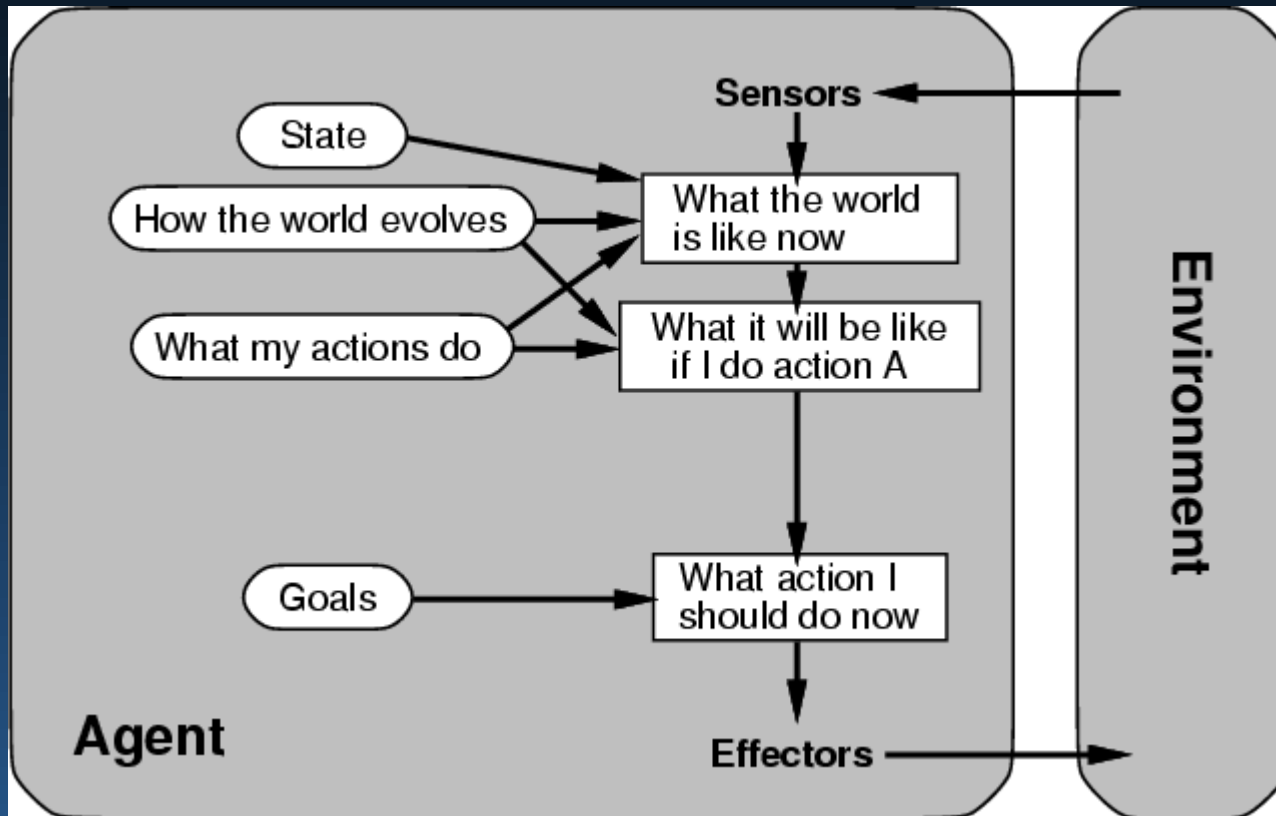
```
function REFLEX-AGENT-WITH-STATE(percept) returns action  
  static: state, a description of the current world state  
           rules, a set of condition-action rules  
  
  state ← UPDATE-STATE(state, percept)  
  rule ← RULE-MATCH(state, rules)  
  action ← RULE-ACTION[rule]  
  state ← UPDATE-STATE(state, action)  
  return action
```

- Stany wewnętrzne pozwalają na zróżnicowanie reakcji przy tych samych stanach percepcyjnych, w zależności od przeszłego działania agenta.
- Wymaga modelu świata, reprezentacji jego zmian i wpływu własnych działań na stan świata.
- Ograniczenie się tylko do poprzedniego stanu znacznie upraszcza budowę agenta (łańcuchy Markowa).

# Agent z celami



Agenci reprezentujący stany wewnętrzne będące modelem środowiska i stany pożądane.



# Agent z celami



Cel: opis sytuacji pożądanej.

Ogólny: osiągnięcie przewagi w grze.

Szczegółowy: dotarcie do planowanego miejsca.

Ostatni stan nie wystarczy, konieczny jest opis stanu oparty na reprezentacji skutków wirtualnych akcji.

Stan wewnętrzny: pozwala śledzić zmiany stanu świata, nie tylko bezpośrednio spostrzegane zmiany.

Rozważanych jest wiele decyzji: co się stanie jeśli ...

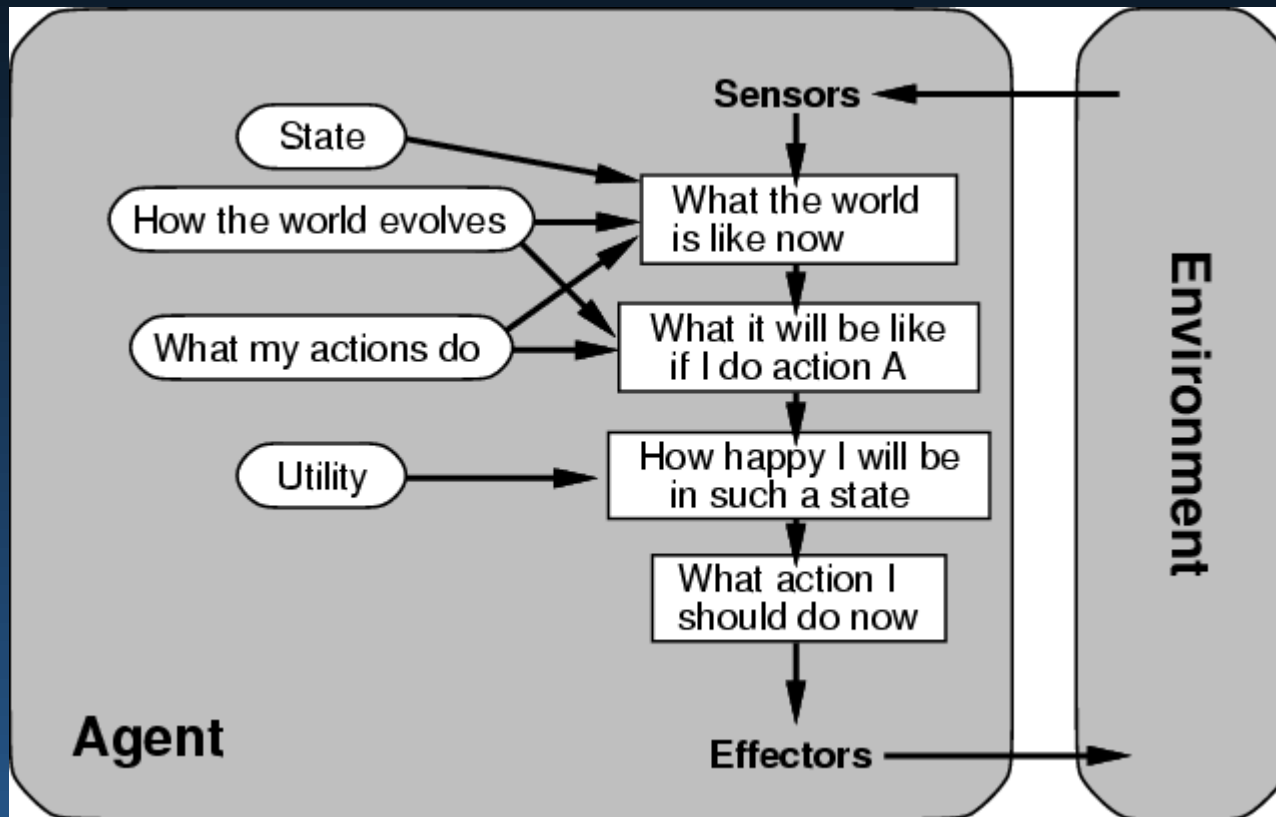
Podstawą działania jest planowanie i szukanie.



# Agent z f. użyteczności



Stany oceniane za pomocą f. użyteczności.



# Agent z f. użyteczności



Preferowane stany wewnętrzne mają wyższe wartości funkcji użyteczności – modeluje to ukierunkowanie emocjonalne.

Np. należy wybrać szybsze, bezpieczniejsze lub pewniejsze środki transportu.

F. użyteczności  $f$ : stan  $\Rightarrow U(\text{stan})$

F. użyteczności umożliwia:

- rozstrzyganie pomiędzy sprzecznymi celami.
- szansą na sukces a wagą celu.

Szukanie – związane z celami, działania bezpośrednie.

Użyteczność – związana z osiągnięciem dalekosiężnych celów (np. wygranej lub zdobyciu przewagi w grach).

# Agent ds. zakupów



Agent krążący po sklepie powinien orientować się w przestrzeni:

- agent refleksyjny – ma szczegółową mapę (przy zmianie położeń ma kłopoty).
- agent celowy – sam tworzy mapę i wykorzystuje ją do rozumowania, dostosowuje się do zmian, np. wyprzedaży.

Wybieranie przedmiotów: musi rozpoznawać przedmioty.

- agent refleksyjny – zbiera przedmioty dobrze wyglądające.
- agent celowy – posługuje się listą zakupów, ocenia wybierane przedmioty. Może planować, np. jeśli nie znajduje poszukiwanego przedmiotu planuje inne sposoby jego zakupu.
- agent z f. użyteczności – uwzględnia jakość i cenę.

# Przykłady agentów



Działają wiele agentów programowych:

- donoszące o błędach w programie – czuwają i przy wykryciu błędu zbierają informacje i poszukują właściwego odbiorcy.
- agent pocztowy – dostarcza pocztę, poszukując odbiorcy.
- agent WWW – poszukuje informacji w sieci korzystając z różnych mechanizmów wyszukiwawczych.
- agent bibliograficzny – czuwa nad aktualizacją danych bibliograficznych, linków WWW.
- agent bioinformatyczny – śledzi pojawianie się nowych informacji i nowych baz informatycznych, dokonuje konwersji formatów danych.
- agent prowadzący samochód: 27.06.2011 Nevada zalegalizowała samochody z automatycznym kierowcą (Google cars).
- Robot Proces Automation (RPA), system reguły automatyzacji schematycznych procesów, pozwala agentom na wykonywanie powtarzalnych czynności.

# Agenci na co dzień

Microsoft Agent technology (od 1997!),  
używający animowanych postaci:



- powiadamia i czyta pocztę elektroniczną;
- zbiera i odczytuje informacje z giełdy, serwisów pogodowych i wybranych stron WWW, np. ostatnie wiadomości, ostrzeżenia przed wirusami;
- dostarcza pozdrowienia i wiadomości odczytujące je lub odśpiewując;
- przypomina o zdarzeniach z kalendarza: spotkaniach, rocznicach etc ;
- opowiada kawały, ściągając je z internetu; konwersja formatów danych;
- znajduje interesujące strony WWW i je poleca.

Obecnie zastąpiła to Cortana i MS Bing, ale bez awatara, bez polskiej wersji (4/2022). Nie ma też polskich wersji Alexy czy Siri i innych agentów.

Telegram Bots pozwala na przesyłanie wiadomości używając botów.

Lista polskich botów.

BotPrize new competition: czy gram z komputerem czy człowiekiem?

# Świat agentów



Każda duża firma IT tworzy swoich agentów, głównie chatboty.

CyberBuddy korzysta z old Microsoft Agent Talking Buddy (tylko historia).

Dodaj swojemu botowi inteligencji z MS Cognitive Services  
Microsoft Azure Bot Service

Alexa Prize Competition 3.5 M\$ nagrody, “advancing Conversational AI”

Ultra Hal Assistant – Hal, agent od Zabaware; podtrzymuje dialogi, pamięta.

Long list of ChatBots narzędzia do tworzenia swoich agentów.

Zbuduj sobie bota: Pandora chatbot. albo Telegram Bot

Polskie boty (stare) <https://chatbot.pl/> Bots.ai (Warta)

Desperados Bot pierwsza loteria w Polsce prowadzona przez chatbota.

BotPrize nagroda 1.000.000 \$.

Digital Twin Applications i technologia cyfrowych bliźniaków MS Azure.

**WD: Świat Bytów Wirtualnych**, wykład (6 godzin) z demonstracjami  
Program (PDF), prezentacja część 1 (PPT) oraz część 2 (PPT) (2005).

# HumanEval



**HumanEval** to zestaw testów do ocen przydatności modeli LLM do pisania programów komputerowych, opisany w publikacji:

Chen et al. [Evaluating Large Language Models Trained on Code](#)

Ten test służy do pomiaru poprawności funkcjonalnej syntezy programów.

Składa się z 164 oryginalnych problemów programistycznych, oceniających rozumienie języka, tworzenie algorytmów i proste zadania matematyczne opisane w sposób werbalny, z niektórymi zadaniami porównywalnymi do prostych pytań zadawanych w czasie egzaminu dla inżynierów programowania przez duże firmy, takie jak Amazon.

Wyniki wielu benchmarków zbierane są przez HuggingFace dla modeli otwartych:

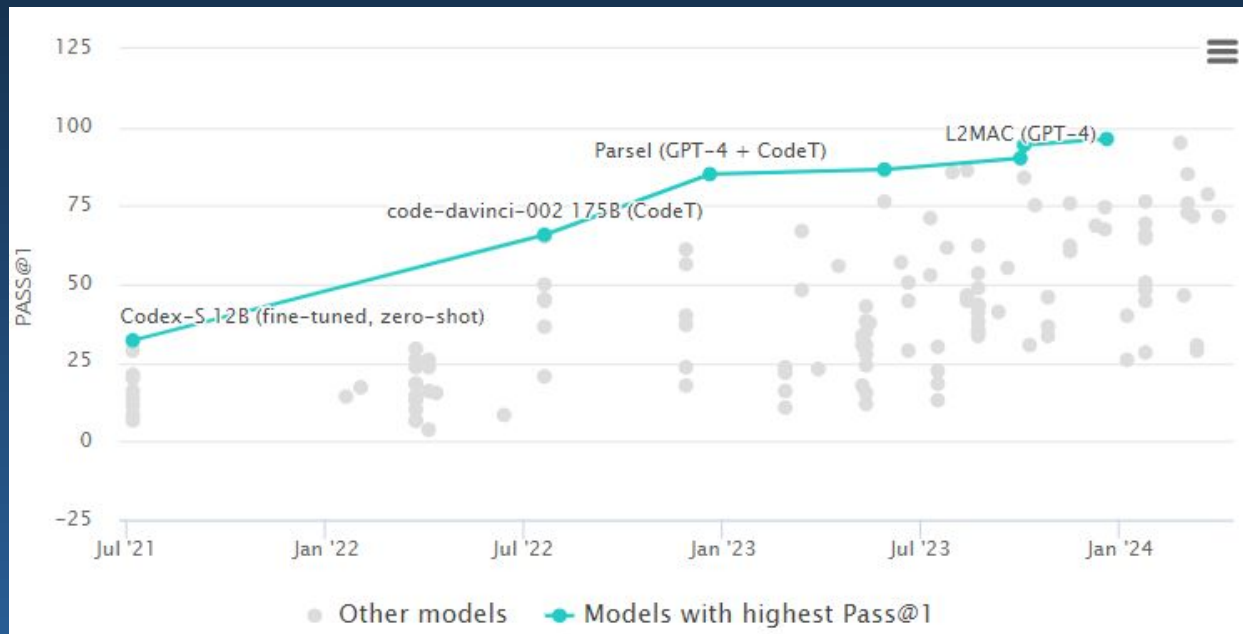
[The Big Benchmarks Collection](#)

# Agenci i modele językowe



**Language Agent Tree Search (LATS)** to uniwersalny framework wykorzystujący Large Language Models (LLMs) do planowania, rozumowania i działania. LATS używa LLMs jako agentów, ocen heurystycznych i optymalizacji funkcji w procesie podejmowania decyzji.

Porównanie AutoDev z Language Agent Tree Search (LATS) i Reflexion, to najlepsze obecnie systemy na testach HumanEval, **HumanEval leaderboard (3/2024)**





# Reprezentacja wiedzy: przykładowe pytania

- Co reprezentują elementy sieci semantycznej?
- Jak prowadzi się rozumowanie dla sieci semantycznych?
- Jak reprezentowane są słowa w mózgu? Jak to się ma do sieci semantycznych?
- W jaki sposób można przewidzieć aktywację mózgu dla nowych pojęć?
- Na czym polega kreatywność przy wymyślaniu nowych słów?
- Co to jest system produkcyjny? Z czego się składa i jak działa?
- Jakie są zalety i wady systemów produkcyjnych?
- Podać przykład ramy dla pojęcia XXX.
- Rodzaje agentów.
- Czy agent typu X wystarczy by zrobić Y?