

Początki Lispu

((Archeologia Cyfrowa)))

Jakub Grobelny

19.11.2019

O czym będzie?

- 1 Do czego był potrzebny taki język?
- 2 „*Recursive Functions of Symbolic Expressions...*”
- 3 Pierwsze implementacje
- 4 Być może coś więcej (Lisp-maszyny, Scheme)

Narodziny sztucznej inteligencji

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

- „Czy maszyny myślą?”

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

- „Czy maszyny myślą?”
- „Czy maszyny mogą myśleć?”

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

- „*Czy maszyny myślą?*”
- „*Czy maszyny mogą myśleć?*”
- „*Czy maszyny mogą działać nieodróżnialnie od ludzi?*”

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

- „*Czy maszyny myślą?*”
- „*Czy maszyny mogą myśleć?*”
- „*Czy maszyny mogą działać nieodróżnialnie od ludzi?*”

1951 Marvin Lee Minsky buduje sieć neuronową SNARC¹

¹Stochastic neural analog reinforcement calculator

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

- „*Czy maszyny myślą?*”
- „*Czy maszyny mogą myśleć?*”
- „*Czy maszyny mogą działać nieodróżnialnie od ludzi?*”

1951 Marvin Lee Minsky buduje sieć neuronową SNARC¹

1951 Pierwsze programy grające w warcaby (Christopher Strachey) i szachy (Dietrich Prinz)

¹Stochastic neural analog reinforcement calculator

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

- „*Czy maszyny myślą?*”
- „*Czy maszyny mogą myśleć?*”
- „*Czy maszyny mogą działać nieodróżnialnie od ludzi?*”

1951 Marvin Lee Minsky buduje sieć neuronową SNARC¹

1951 Pierwsze programy grające w warcaby (Christopher Strachey) i szachy (Dietrich Prinz)

¹Stochastic neural analog reinforcement calculator

Narodziny sztucznej inteligencji

Lata 50. XX wieku były czasem, gdy sztuczna inteligencja pojawiła się jako dziedzina wiedzy.

1950 Alan Turing publikuje „*Computing Machinery and Intelligence*”

- „*Czy maszyny myślą?*”
- „*Czy maszyny mogą myśleć?*”
- „*Czy maszyny mogą działać nieodróżnialnie od ludzi?*”

1951 Marvin Lee Minsky buduje sieć neuronową SNARC¹

1951 Pierwsze programy grające w warcaby (Christopher Strachey) i szachy (Dietrich Prinz)

¹Stochastic neural analog reinforcement calculator

Narodziny sztucznej inteligencji

1955 Allen Newell, Herbert A. Simon i Cliff Shaw tworzą program „*Logic Theorist*”

Narodziny sztucznej inteligencji

- 1955 Allen Newell, Herbert A. Simon i Cliff Shaw tworzą program „*Logic Theorist*”
- Program naśladujący ludzkie techniki rozwiązywania problemów

Narodziny sztucznej inteligencji

1955 Allen Newell, Herbert A. Simon i Cliff Shaw tworzą program „*Logic Theorist*”

- Program naśladowujący ludzkie techniki rozwiązywania problemów
- Program manipulujący **symbolami**

²Autorstwa Alfreda Northa Whiteheada i Bertranda Russella

Narodziny sztucznej inteligencji

1955 Allen Newell, Herbert A. Simon i Cliff Shaw tworzą program „*Logic Theorist*”

- Program naśladowujący ludzkie techniki rozwiązywania problemów
- Program manipulujący **symbolami**
- Udowodnił 38 z pierwszych 52 twierdzeń z „*Principia Mathematica*”²

²Autorstwa Alfreda Northa Whiteheada i Bertranda Russella

Narodziny sztucznej inteligencji

1955 Allen Newell, Herbert A. Simon i Cliff Shaw tworzą program „*Logic Theorist*”

- Program naśladowujący ludzkie techniki rozwiązywania problemów
- Program manipulujący **symbolami**
- Udowodnił 38 z pierwszych 52 twierdzeń z „*Principia Mathematica*”² (niektóre z dowodów były bardziej eleganckie niż wcześniej istniejące)...

²Autorstwa Alfreda Northa Whiteheada i Bertranda Russella

Narodziny sztucznej inteligencji

1955 Allen Newell, Herbert A. Simon i Cliff Shaw tworzą program „*Logic Theorist*”

- Program naśladowujący ludzkie techniki rozwiązywania problemów
- Program manipulujący **symbolami**
- Udowodnił 38 z pierwszych 52 twierdzeń z „*Principia Mathematica*”² (niektóre z dowodów były bardziej eleganckie niż wcześniej istniejące)...
- ...i to wszystko zanim określenie „sztuczna inteligencja” w ogóle zostało stworzone.

²Autorstwa Alfreda Northa Whiteheada i Bertranda Russella

Narodziny sztucznej inteligencji

1956 Konferencja w
Dartmouth.

Narodziny sztucznej inteligencji

1956 Konferencja w
Dartmouth. **John
McCarthy** przekonuje
zebranych do używania
terminu „*sztuczna
inteligencja*”.



Rysunek: John McCarthy

Symbole

Symbole

Okazuje się, że operowanie na **symbolach** jest istotne przy tworzeniu sztucznej inteligencji.

Symbole

Okazuje się, że operowanie na **symbolach** jest istotne przy tworzeniu sztucznej inteligencji.

Ludzkie rozumowanie opiera się na manipulacji symbolami (*physical symbol system hypothesis* – Allan Newell i Herbert A. Simon)

Symbole

Okazuje się, że operowanie na **symbolach** jest istotne przy tworzeniu sztucznej inteligencji.

Ludzkie rozumowanie opiera się na manipulacji symbolami (*physical symbol system hypothesis* – Allan Newell i Herbert A. Simon)

System symboli składa się z symboli, składania ich w struktury (wyrażenia) i manipulowania nimi (przetwarzania) w celu tworzenia nowych wyrażień.

Symbole

W 1959 roku John McCarthy pisze pracę „*Programs with common sense*”.

Symbole

W 1959 roku John McCarthy pisze pracę „*Programs with common sense*”.

Proponuje w niej stworzenie programu „*advice taker*”, który rozwiązywałby problemy poprzez manipulację zdaniami (**symbole!**).

Symbole

W 1959 roku John McCarthy pisze pracę „*Programs with common sense*”.

Proponuje w niej stworzenie programu „*advice taker*”, który rozwiązywałby problemy poprzez manipulację zdaniami (**symbole!**).

„*Our ultimate objective is to make programs that learn from their experience as effectively as humans do.*”

„Programs with common sense”

„*Programs with common sense*”

„*A class of entities called terms is defined and a term is an expression. A sequence of expressions is an expression. These expressions are represented in the machine by list structures*” – reprezentacja programów w postaci **list symboli**

Przykłady:

Przykłady:

at(I, desk)

at(desk, home)

at(car, home)

at(home, county)

at(airport, county)

Przykłady:

at(I, desk)

at(desk, home)

at(car, home)

at(home, county)

at(airport, county)

$at(x, y), at(y, z) \rightarrow at(x, z)$

Przykłady:

$at(I, desk)$

$at(desk, home)$

$at(car, home)$

$at(home, county)$

$at(airport, county)$

$at(x, y), at(y, z) \rightarrow at(x, z)$

$transitive(at)$

$transitive(u) \rightarrow (u(x, y), u(y, z) \rightarrow u(x, z))$

Przykłady:

at(I, desk)

at(desk, home)

at(car, home)

at(home, county)

at(airport, county)

$at(x, y), at(y, z) \rightarrow at(x, z)$

transitive(at)

$transitive(u) \rightarrow (u(x, y), u(y, z) \rightarrow u(x, z))$

Prawie jak Prolog?

Zapotrzebowanie na nowe języki

Information Processing Language – niskopozimowy język programowania do manipulowania listami stworzony przez Newella, Shawa i Simona, który posłużył do napisania programu „*Logic Theorist*”.

IPL-V List

Structure

Example

Name	SYMB	LINK
L1	9-1	100
100	S4	101
101	S5	0
9-1	0	200
200	A1	201
201	V1	202
202	A2	203
203	V2	0

Rysunek: Kod w IPL

