|  |
| --- |
| Wydział Matematyki i Nauk Informacyjnych |
| Sprawozdanie |
| Identyfikacja chmur |
|  |
| **Łukasz Dragan** |
| **2016-12-04** |

# Kod programu

Kod programu dzieli się na dwa pliki:

1. main.pl: jądro systemu zarządzające przeprowadzeniem identyfikacji chmur.

cloud(X) :-

retractall(known(Y,A)),

consult('db.pl'),

cloudBuf(X).

:- dynamic known/2.

ask(A, V):-

known(V, A),

!.

ask(A, V):-

known(\_, A),

!,

fail.

ask(A, V):-

known(V2, A),

V \== V2,

!,

fail.

ask(A, V):-

write(A),

write('? : '),

read(Y),

asserta(known(Y, A)),

Y == V.

save\_fact(Cloud,FeaturesList, NonFeaturesList) :-

writeEx(["\ncloudBuf("]),

writeEx([Cloud]),

writeEx([") :-"]),

writeEx2(FeaturesList, y),

writeEx2(NonFeaturesList, n),

writeEx(["!."]).

writeEx([]).

writeEx([H|T]):-

open('db.pl',append,Stream),

write(Stream, H),

close(Stream),

writeEx(T).

writeEx2([], A).

writeEx2([H|T], A):-

open('db.pl',append,Stream),

write(Stream, 'ask('),

write(Stream, H),

write(Stream, ', '),

write(Stream, A),

write(Stream, '),'),

nl(Stream),

close(Stream),

writeEx2(T, A).

1. db.pl: dynamiczna baza wiedzy. Plik zawiera reguły, wg których system, zadając pytania, rozpoznaje chmurę.

cloudBuf(nimbostratus):-

ask(pada\_deszcz, y),

ask(zaslania\_cale\_niebo, y).

cloudBuf(cumulonimbus):-

ask(pada\_deszcz, y).

cloudBuf(stratus) :-

ask(jest\_nisko, y),

ask(zaslania\_cale\_niebo, y),

ask(jest\_zwarta, y).

cloudBuf(stratocumulus) :-

ask(jest\_nisko, y),

ask(zaslania\_cale\_niebo, y).

cloudBuf(altostratus) :-

ask(jest\_gladka, y),

ask(pokrywa\_wiekszosc\_nieba, y).

cloudBuf(cirrostratus) :-

ask(jest\_gladka, y),

ask(jest\_zwarta, y).

cloudBuf(cirrus) :-

ask(jest\_gladka, y).

cloudBuf(altocumulus) :-

ask(pokrywa\_wiekszosc\_nieba, y).

# Funkcjonalności i przykładowe wywołania

W skład wchodzą dwie funkcjonalności:

1. Rozpoznanie chmury.

Następuje za pomocą komendy cloud(X). , po której system zadaje pytania niezbędne do rozpoznania chmury. Możliwe odpowiedzi na pytania to: „y.” lub „n.”. Jeżeli w bazie wiedzy znajduje się chmura opisana odpowiedziami na zadane pytanie.

Przykładowe wywołanie:

1 ?- cloud(X).

pada\_deszcz? : n.

jest\_nisko? : |: y.

zaslania\_cale\_niebo? : |: y.

jest\_zwarta? : |: y.

X = stratus .

Szukaną chmurą okazał się stratus.

1. Rozszerzanie bazy wiedzy.

Za pomocą funkcji sace\_fact/3 można rozszerzać bazę wiedzy programu. Do pliku db.pl zostaje wtedy dopisana odpowiednia reguła. Pierwszy argument to nazwa zjawiska. Drugi: zestaw faktów które muszą zajść, aby zaistniało dane zjawisko. Trzeci: zestaw faktów, które nie mogą zająć, aby zjawisko zaistniało.

Przykładowe wywołanie:

1 ?- save\_fact(mgla,[slaba\_widocznosc],[pada\_deszcz]).

true.

2 ?- cloud(X).

pada\_deszcz? : n.

jest\_nisko? : |: n.

jest\_gladka? : |: n.

pokrywa\_wiekszosc\_nieba? : |: n.

slaba\_widocznosc? : |: y.

X = mgla.