

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).
```

2. 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 wchodzi 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.

2. 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 zajść, 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.
```