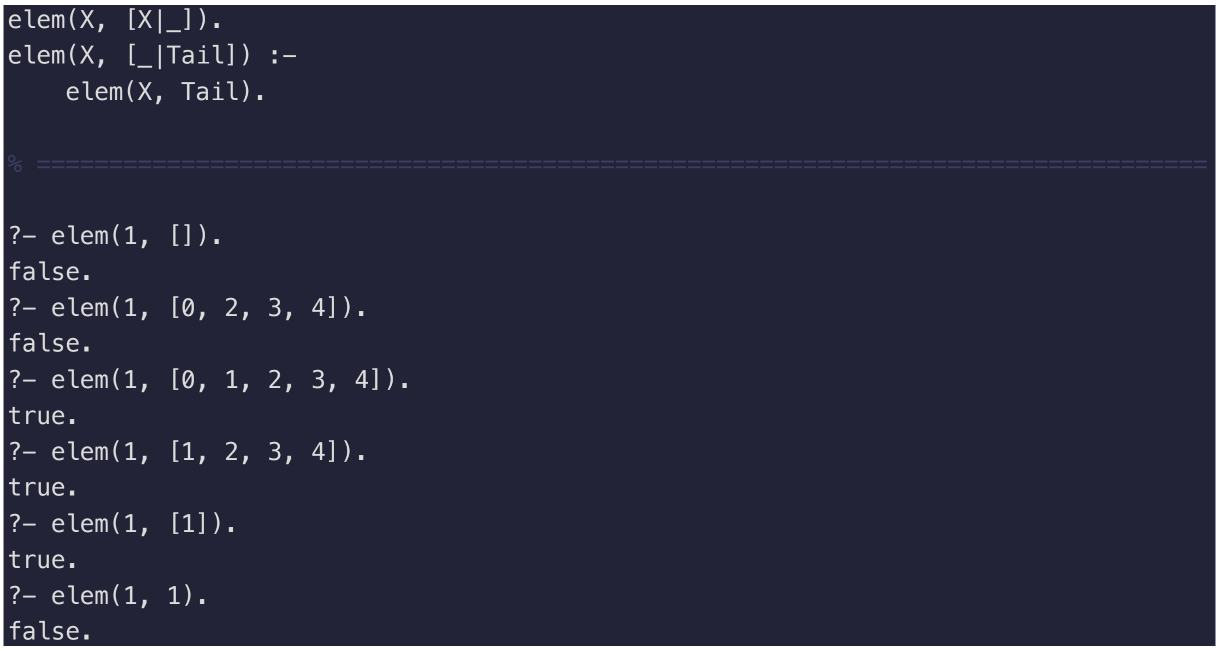
Inżynieria wiedzy i symboliczne uczenie maszynowe

Laboratorium 4 — Prolog

Jakub Szaredko

Ćwiczenie 1

`elem` sprawdzający czy element należy do listy.



```
elem(X, [X|_]).
elem(X, [_|Tail]) :-
    elem(X, Tail).

% =================================================================================

?- elem(1, []).
false.
?- elem(1, [0, 2, 3, 4]).
false.
?- elem(1, [0, 1, 2, 3, 4]).
true.
?- elem(1, [1, 2, 3, 4]).
true.
?- elem(1, [1]).
true.
?- elem(1, 1).
false.
```

Ćwiczenie 2

`l` zwracający długość listy.

```
l([], 0).
l([_|Tail], N) :-
    l(Tail, N1),
    N is N1 + 1.

% ================================================================================


?- l([], N).
N = 0.
?- l([0, 1, 2, 3, 4], N).
N = 5.
?- l(1, N).
false.
```

Ćwiczenie 3

Zaimplementuj prosty system ekspertowy, wnioskowanie powinno opierać się o co najmniej 5 reguł i 10 faktów.

Zaimplementowany został system ekspertowy szacujący prędkość ([km/h]) poruszania się entuzjasty turystyki górskiej lub grupy entuzjastów. System jest oparty na 11 faktach w dynamicznej bazie wiedzy oraz na 6 lub 7 (jeśli wliczone zostanie `init_example` 😝) regułach.

```
:- dynamic
    terrain/1,            % flat | moderate | steep
    trail_condition/1,    % dry | muddy | icy
    weather/1,            % clear | cloudy | rain | storm
    temperature/1,        % Temperature in Celsius degrees
    wind_speed/1,         % none | light | moderate | strong
    altitude_change/1,    % low | medium | high
    hiker_experience/1,   % beginner | intermediate | experienced
    group_size/1,         % Number of people in the group
    backpack_weight/1,    % Weight in kilograms
    is_trail_marked/0,    % true if trail is marked
    footwear/1.           % proper | improper

speed_value(high, S) :- random_between(50, 65, R), S is R / 10.
speed_value(medium, S) :- random_between(30, 49, R), S is R / 10.
speed_value(low, S) :- random_between(20, 29, R), S is R / 10.
speed_value(very_low, S) :- random_between(10, 19, R), S is R / 10.

slower_than(A, B) :- A = high, B = medium.
slower_than(A, B) :- A = medium, B = low.
slower_than(A, B) :- A = low, B = very_low.
slower_than(A, B) :- A = very_low, B = very_low.

is_weather_fine :-
    weather(clear)
    ; weather(cloudy).

base_speed(high) :-
    terrain(flat),
    trail_condition(dry),
    weather(clear),
    hiker_experience(experienced),
    altitude_change(low).

base_speed(medium) :-
    terrain(moderate),
    trail_condition(dry),
    weather(cloudy),
    hiker_experience(intermediate),
    altitude_change(medium).

base_speed(low) :-
    terrain(steep)
    ; trail_condition(muddy)
    ; weather(rain)
    ; altitude_change(high).

base_speed(very_low) :-
    weather(storm);
    trail_condition(icy);
    \+ is_trail_marked.

base_speed(very_low) :-
    temperature(T), T < 5.

base_speed(very_low) :-
    wind_speed(strong).

base_speed(very_low) :-
    backpack_weight(W), W >= 15.

base_speed(low) :-
    backpack_weight(W), W >= 10, W < 15.

base_speed(low) :-
    wind_speed(moderate).

base_speed(low) :-
    temperature(T), T >= 5, T =< 10.

base_speed(low) :-
```

```prolog
    footwear(improper).

base_speed(medium) :-
    is_weather_fine.

estimate_speed(SpeedKmH) :-
    base_speed(BaseSpeed),
    group_size(N),
    (
        N > 4 -> slower_than(BaseSpeed, AdjustedSpeed)
        ; AdjustedSpeed = BaseSpeed
    ),
    speed_value(AdjustedSpeed, SpeedKmH).

estimate_time(DistanceKm, TimeH) :-
    estimate_speed(Speed),
    TimeH is DistanceKm / Speed.

init_example :-
    retractall(terrain(_)),
    retractall(trail_condition(_)),
    retractall(weather(_)),
    retractall(temperature(_)),
    retractall(wind_speed(_)),
    retractall(altitude_change(_)),
    retractall(hiker_experience(_)),
    retractall(group_size(_)),
    retractall(backpack_weight(_)),
    retractall(footwear(_)),
    retractall(is_trail_marked),

    assert(terrain(moderate)),
    assert(trail_condition(muddy)),
    assert(weather(cloudy)),
    assert(temperature(10)),
    assert(wind_speed(moderate)),
    assert(altitude_change(medium)),
    assert(hiker_experience(intermediate)),
    assert(group_size(5)),
    assert(backpack_weight(5)),
    assert(footwear(proper)),
    assert(is_trail_marked).

% ================================================================================

?- init_example.
true.
?- estimate_speed(S).
S = 1.1
```