

Petr Jeniš

Výběr dopravního prostředku

—

program v jazyce Prolog

**Výstupní práce
pro předmět
Umělá inteligence (UIN)
2011/2012**



TECHNICKÁ UNIVERZITA V LIBERCI
www.tul.cz



Účel programu:

Program se pokouší uhodnout, jaký dopravní prostředek z těchto (chůze pěšky, bicykl, motocykl, letadlo, plachetnice, solární automobil, ponorka, elektromobil a neznámý) má uživatel nejspíše na mysli. Rozhoduje se na základě otázek s možnými odpověďmi ano/ne. Otázky zobrazuje na základě pravidel jak pro každý dopravní prostředek zvlášť, tak pro pravidla skupin, do kterých jsou dopravní prostředky zařazeny. Jedná se o skupiny: pohon lidskou silou, spalovacím motorem, přírodními silami a elektřinou. Program je napsaný v jazyce **SWI-Prolog**, jehož interpreter a editor je možno zdarma stáhnout na síti Internet [1].

Ovládaní programu:

V zásadě velmi jednoduché, po načtení zdrojového kódu a spuštění zadáme "dotaz" `?-zacni.` a program se spustí. Na otázky odpovídáme `ano.` nebo `ne.`, případně `a.` nebo `n.` Po skončení je možné opět použít příkaz `?-zacni.` Je jen potřeba dodržet syntaxi a nezapomenout psát tečky za příkazy.

Popis funkce programu:

Po spuštění procedury "`zacni`" program skočí do otázkové části, protože následuje jako první za částmi, které jsou v podstatě definici proměnných. Procedura "`zacni`" sice obsahuje zdrojový kód, ale ten je nyní z uživatelského hlediska bezvýznamný, protože nedošlo k naplnění proměnné „`Prostředek`“ žádným výstupem. Vypíše nám tedy otázku a čeká na vstup od uživatele.

Pokud je položena otázka „`o`“ a odpověď je "ano", tak se odpověď zaznamená jako "`ano(o)`" a úspěchem, v opačném případě je odpověď zaznamenána jako "`ne(o)`" a následována neúspěchem. Kladné odpovědi je nutno uschovávat, protože kdyby po nich následovala odpověď negativní (na tutéž otázku) tak bychom cíle nikdy nedosáhli. Pozitivní odpověď na tutéž otázku logicky způsobí, že hypotéza není vyvrácena a může vyústit v úspěch (konkrétně jiné vozidlo ve stejné kategorii pohonu). Tímto způsobem se program vyvaruje opakování té samé otázky. Následná metoda ověření podmínek otázky „`o`“ je tato: Pokud se v paměti nachází "`ano(o)`", jedna se o úspěch a pokračování, pokud se v paměti nachází "`ne(o)`", jedna se o neúspěch a pokud se v paměti nenachází nic z výše uvedeného, polož otázku „`o`“. Tento cyklus se opakuje, dokud nejsou splněna pravidla pro některý dopravní prostředek a jeho skupinu, eventuálně "projde" až na konec a dopravní prostředek je zůstane neznámý.

Hlavní využití příkazy: assert a retract

Kazdy program v prologu je složen z množiny relací, všechny relace můžeme považovat za relační databázi [2].

Příkaz `assert(x)` přidává do databáze záznam vázaný na proměnnou `X`. Cil `X` je možné splnit pouze jednou a není možné ho při navracení použít znovu. V našem případě je tedy nutno použít na konci příkaz `retract` a vymazat databázi (viz procedura „`vycistit`“)

Příkaz `assert` ovšem funguje pouze v případě, že procedury jsou definovány jako dynamické. Toto je v programu ošetřeno v "otázkové části" v řádku `:- dynamic ano/1, ne/1.`

obr. 1 – zdrojový kód programu

```

/* UVODNI CAST PROGRAMU */
zacni :- uvazuje(Prostrek),
        write('Myslim ze vas dopravní prostředek je: '),
        write(Prostrek),
        nl,
        vycistit.

/* TESTOVANE MOZNOSTI */
uvazuje(chuze_pesky)      :- chuze_pesky, !.
uvazuje(bicykl)           :- bicykl, !.
uvazuje(motocykl)         :- motocykl, !.
uvazuje(letadlo)          :- letadlo, !.
uvazuje(plachetnice)      :- plachetnice, !.
uvazuje(solarni_automobil) :- solarni_automobil, !.
uvazuje(ponorka)          :- ponorka, !.
uvazuje(elektromobil)     :- elektromobil, !.
uvazuje(neznamy).

/* PRAVIDLA PRO ROZPOZNANI DOPRAVNIHO PROSTREDKU */
chuze_pesky      :- lidskypohon,
                  prirodnisily,
                  overime(nepotrebujeme_pomucky),
                  overime(vyzaduje_pevnou_obuv).

bicykl           :- lidskypohon,
                  overime(ma_dve_kola),
                  overime(ma_slapky).

motocykl         :- spalovacimotor,
                  overime(pohybuje_se_rychle),
                  overime(casto_zavodni_prostrek).

letadlo          :- spalovacimotor,
                  overime(ma_kridla),
                  overime(leta),
                  overime(pohybuje_se_rychle).

plachetnice      :- prirodnisily,
                  overime(neleta),
                  overime(plave).

solarni_automobil :- prirodnisily,
                  overime(neleta),
                  overime(neplave),
                  overime(ma_uzka_kola),
                  overime(funguje_jen_kdyz_sviti_slunce).

ponorka          :- elektrina,
                  overime(plave),
                  overime(neleta),
                  overime(obvykle_neni_videt).

elektromobil     :- elektrina,
                  overime(ma_kola),
                  overime(neprodukuje_zplodiny).

/* KLASIFIKACE DO TRID */
lidskypohon      :- overime(pohanene_clovekem), !.
lidskypohon      :- overime(nizka_hmotnost).

spalovacimotor   :- overime(potrebuje_benzin), !.
spalovacimotor   :- overime(produkuje_zplodiny),
                  overime(pomerne_hlucne).

prirodnisily     :- overime(neobvykly_zdroj_hybne_sily), !.
prirodnisily     :- overime(obnovitelny_zdroj_energie),
                  overime(neni_treba_dobijet),
                  overime(neni_treba_tankovat).

elektrina        :- overime(nema_spalovaci_motor), !.
elektrina        :- overime(ma_baterie).

/* OTAZKOVA CAST */
ask(Otazka) :-
    write('Ma vas dopravní prostředek nasledující vlastnost: '),
    write(Otazka),
    write('? '),
    read(Odpoved),
    nl,
    ( (Odpoved == ano ; Odpoved == a)
      ->
        assert(ano(Otazka)) ;
        assert(ne(Otazka)), fail).

:- dynamic ano/1, ne/1.

/* OVEROVACI CAST */
overime(Stav) :-
    (ano(Stav)
      ->
        true ;
    (ne(Stav)
      ->
        fail ;
    ask(Stav))).

/* VYCISTENI PROMENNYCH PRED PRIPADNYM DALSIM PRUCHODEM PROGRAMU */
vycistit :- retract(ano(_)), fail.
vycistit :- retract(ne(_)), fail.
vycistit.

```

Příklad – dva ukázkové průchody programem

- Spouštěno v programu SWI-Prolog 5.10.5 [2]

1 ?- zacni.

Ma vas dopravní prostředek nasledující vlastnost: pohanene_clovekem? ne.

Ma vas dopravní prostředek nasledující vlastnost: nizka_hmotnost? ne.

Ma vas dopravní prostředek nasledující vlastnost: potrebuje_benzin? ano.

Ma vas dopravní prostředek nasledující vlastnost: pohybuje_se_rychle? ano.

Ma vas dopravní prostředek nasledující vlastnost: casto_zavodni_prostredok? ne.

Ma vas dopravní prostředek nasledující vlastnost: ma_kridla? ano.

Ma vas dopravní prostředek nasledující vlastnost: leta? ano.

Myslím že vas dopravní prostředek je: letadlo
true.

2 ?- zacni.

Ma vas dopravní prostředek nasledující vlastnost: pohanene_clovekem? ne.

Ma vas dopravní prostředek nasledující vlastnost: nizka_hmotnost? ano.

Ma vas dopravní prostředek nasledující vlastnost: neobvykly_zdroj_hybne_sily? ne.

Ma vas dopravní prostředek nasledující vlastnost: obnovitelny_zdroj_energie? ano.

Ma vas dopravní prostředek nasledující vlastnost: neni_treba_dobijet? ano.

Ma vas dopravní prostředek nasledující vlastnost: neni_treba_tankovat? ano.

Ma vas dopravní prostředek nasledující vlastnost: nepotrebujeme_pomucky? ano.

Ma vas dopravní prostředek nasledující vlastnost: vyzaduje_pevnou_obuv? ne.

Ma vas dopravní prostředek nasledující vlastnost: ma_dve_kola? ne.

Ma vas dopravní prostředek nasledující vlastnost: potrebuje_benzin? ne.

Ma vas dopravní prostředek nasledující vlastnost: produkuje_zplodiny? ne.

Ma vas dopravní prostředek nasledující vlastnost: neleta? ano.

Ma vas dopravní prostředek nasledující vlastnost: plave? ne.

Ma vas dopravní prostředek nasledující vlastnost: neplave? ano.

Ma vas dopravní prostředek nasledující vlastnost: ma_uzka_kola? ano.

Ma vas dopravní prostředek nasledující vlastnost: funguje_jen_kdyz_sviti_slunce?
ano.

Myslím že vas dopravní prostředek je: solarni_automobil
true.

Reference:

- [1] <http://www.swi-prolog.org/download/stable>
- [2] <http://bulba.sdsu.edu/prolog/parsing/>