

Kurs języka Prolog 2019

Lista zadań nr 1

Na zajęcia 6 marca 2019

Cztery zadania z listy 0 zostały powtórzone, gdyż nie zdążyliśmy ich rozwiązać w zeszłym tygodniu.

Zadanie 1 (1 pkt). Oto jeden z najsłynniejszych sylogizmów:

Wszyscy ludzie są śmiertelni.

Sokrates jest człowiekiem.

Zatem Sokrates jest śmiertelny.

Sformalizuj jego przesłanki w postaci klauzul prologowych i zadam Prologowi pytanie, czy konkluzja jest prawdziwa.

Zadanie 2 (2 pkt). Dane są predykaty:

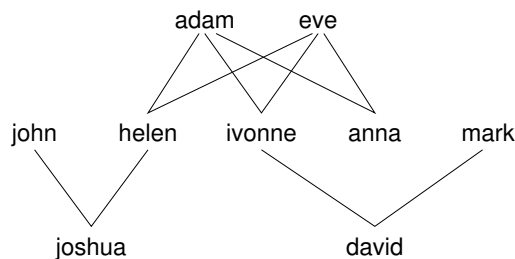
`parent/2` `male/1` `female/1`

Cel `parent(a, b)` jest spełniony wówczas, gdy a jest rodzicem b , zaś cele `male(a)` i `female(a)` — gdy a jest (odpowiednio) mężczyzną bądź kobietą. Korzystając z powyższych predykatów zdefiniuj predykaty:

`sibling/2` `sister/2` `grandson/2` `cousin/2` `descendant/2` `is_mother/1` `is_father/1`

Cel `sibling(a, b)` jest spełniony wówczas, gdy a i b są rodzeństwem, `sister(a, b)` — gdy a jest siostrą b , `grandson(a, b)` — gdy a jest wnukiem b , `cousin(a, b)` — gdy a jest kuzynem b (tj. synem ciotki, wuja lub stryja b), `descendant(a, b)` — gdy a jest potomkiem b , `is_mother(a)` — gdy a jest matką, `is_father(a)` — gdy a jest ojcem.

Dołącz do programu zbiór faktów definiujących predykaty `parent`, `male` i `female` dla następujących zależności rodzinnych:



Zadaj Prologowi następujące pytania:

1. Czy John jest potomkiem Marka?
2. Kto jest potomkiem Adama?
3. Kto jest siostrą Ivonne?
4. Kto ma w tej rodzinie kuzyna i kim ten kuzyn jest?

Narysuj prologowe drzewa poszukiwań dla powyższych zapytań.

Zadanie 3 (2 pkt). Zbuduj prologową bazę danych o bezpośrednich połączeniach kolejowych między miastami (fakt, że istnieje połączenie z miasta A do miasta B nie implikuje, że istnieje połączenie odwrotne):

z	do
wroclaw	warszawa
wroclaw	krakow
wroclaw	szczecin
szczecin	lublin
szczecin	gniezno
warszawa	katowice
gniezno	gliwice
lublin	gliwice

Zapytaj maszynę prologową:

1. czy istnieje bezpośrednie połączenie z Wrocławia do Lublina?
2. z jakimi miastami ma Wrocław bezpośrednie połączenie?
3. z jakich miast można dojechać do Gliwic z dokładnie jedną przesiadką?
4. z jakich miast można dojechać do Gliwic z co najwyżej dwiema przesiadkami? Czemu niektóre miasta są wymienione więcej niż raz?

Na wzór predykatu `descendant/2` z poprzedniego zadania zaprogramuj predykat `connection/2` spełniony wówczas, gdy istnieje połączenie pomiędzy podanymi miastami z dowolną liczbą przesiadek. Wyjaśnij dlaczego Twój predykat może działać niezgodnie z oczekiwaniem (nie musisz umieć zaprogramować poprawnego predykatu — powinienesz jedynie wyjaśnić, czemu taka naiwna implementacja jest niedobra).

Zadanie 4 (2 pkt). Rozważmy predykat `append/3`:

```
append([], X, X).
append([H|T], X, [H|Y]) :-
    append(T, X, Y).
```

Narysuj prologowe drzewo poszukiwań dla celu:

```
?- append(X, Y, [a,b,c]).
```

Zadanie 5 (2 pkt). Rozważmy predykat `select/3`:

```
select(H, [H|T], T).
select(X, [H|T], [H|S]) :-
    select(X, T, S).
```

Narysuj prologowe drzewo poszukiwań dla celu:

```
?- select(x, L, [a,b,c]).
```

Zadanie 6 (2 pkt). Zadać Prologowi następujące pytania:

1. Jakie pary list mają tę własność, że druga jest wynikiem konkatencji dwóch kopii pierwszej z nich?
2. Jaki element należy usunąć z listy `[a,b,c,d]` by otrzymać listę `[a,c,d]`?
3. Jaką listę należy dołączyć do listy `[a,b,c]`, by otrzymać listę `[a,b,c,d,e]`?

Zadanie 7 (3 pkt). Zaprogramuj w Prologu poniższe predykaty:

1. `even(X)`, spełniony gdy lista `X` ma parzystą liczbę elementów;
2. `palindrom(X)`, spełniony gdy lista `X` jest palindromem;
3. `singleton(L)`, spełniony gdy `X` jest listą jednoelementową.
4. `head(H, L)`, spełniony gdy `H` jest pierwszym elementem (głową) listy `L`;
5. `last(L, H)`, spełniony gdy `H` jest ostatnim elementem listy `L`;
6. `tail(T, L)`, spełniony gdy `T` jest listą wszystkich z wyjątkiem pierwszego elementów (ogonem) listy `L`;
7. `init(L, T)`, spełniony gdy `T` jest listą wszystkich z wyjątkiem ostatniego elementów listy `L`;
8. `prefix(P, L)`, spełniony gdy `P` jest listą początkowych elementów (prefiksem) listy `L`;
9. `suffix(L, S)`, spełniony gdy `S` jest listą końcowych elementów (sufiksem) listy `L`.

Zadanie 8 (3 pkt). Skorzystaj z Prologa by wskazać, które ze zdań poniżej są prawdziwe, a które fałszywe.

1. Wszystkie poniższe zdania są prawdziwe.
2. Wszystkie poniższe zdania są fałszywe.
3. Wszystkie powyższe zdania są prawdziwe.
4. Co najmniej jedno z powyższych zdań jest prawdziwe.
5. Wszystkie powyższe zdania są fałszywe.
6. Wszystkie powyższe zdania są fałszywe.

Skorzystaj tylko z atomów `f/0` i `t/0` oznaczających, odpowiednio, fałsz i prawdę, i zdefiniuj predykaty opisujące prawdziwość powyższych zdań. Nie trzeba korzystać z list ani innych nietrywialnych konstrukcji prologowych (np. negacji). Bardzo proste rozwiązanie nie przypomina normalnego programu, ale jest zwarte i elementarne.

Zadanie 9 (3 pkt). Rozwiąż w Prologu poniższą łamigłówkę.

1. Pięć kolorowych domów stoi w rzędzie. Każdy dom ma właściciela, który posiada zwierzę, ulubiony gatunek papierosów i ulubiony napój.
2. Anglik mieszka w czerwonym domu.
3. Hiszpan ma psa.
4. W zielonym domu piją kawę.
5. Rosjanin pije herbatę.
6. Zielony dom sąsiaduje z białym.
7. Właściciel węża pali Winstony.
8. W żółtym domu palą Koole.
9. W domu znajdującym się pośrodku piją mleko.
10. Norweg mieszka w pierwszym domu od lewej.
11. Palacz Chesterfieldów jest sąsiadem właściciela lisa.
12. W domu sąsiadującym z domem właściciela konia palą Koole.
13. Palacz Lucky Strików pije sok.
14. Japończyk pali Kenty.
15. Norweg sąsiaduje z niebieskim domem.

Kto jest właścicielem słonia, a kto pije wódkę? Uwaga: nie używaj list. Nie są potrzebne do rozwiązania tego zadania.