Zadanie zaliczeniowe z Prologu, 2017/18

23 maja 2018

1 Wstęp

1.1 AE-grafy

AE-graf to trójka $\langle V_A, V_E, R\rangle$, gdzie V_A, V_E są dowolnymi rozłącznymi zbiorami ($V_A \cap V_E = \emptyset$) zwanymi zbiorami, odpowiednio, *A-wierzchołków* i *E-wierzchołków*, zaś $R \subseteq (V_A \cup V_E)^2$ jest zwany zbiorem *krawędzi*.

Powiemy, że graf skierowany $\langle V, S\rangle$, gdzie $V = V_A \cup V_E$, zaś $S \subseteq R$ jest *wyborem* z AE-grafu $\langle V_A, V_E, R\rangle$, gdy

- dla każdego $v \in V_A$, i każdego $\langle v, v'\rangle \in R$ zachodzi $\langle v, v'\rangle \in S$,
- dla każdego $v \in V_E$, jeśli istnieje $\langle v, v'\rangle \in R$ dla pewnego v', to istnieje dokładnie jedno $\langle v, v''\rangle \in R$, takie że $\langle v, v''\rangle \in S$.

Powiemy, że lista l jest *listą alternującego przejścia DFS* przez AE-graf $\langle V_A, V_E, R\rangle$, jeśli jest listą wierzchołków odwiedzanych przez algorytm DFS w pewnym grafie będącym wyborem z $\langle V_A, V_E, R\rangle$.

1.2 Reprezentacja AE-grafów

W Prologu AE-grafy (a także wybory z AE-grafów) można reprezentować za pomocą list sąsiedztwa. Reprezentację wierzchołka $v \in V_A$ (oraz $v \in V_E$) stanowi lista `[v, a, v1, ..., vn]` (analogicznie lista `[v, e, v1, ..., vn]`) , gdzie `v` to dowolny unikalny atom będący identyfikatorem wierzchołka, `a` to konkretny atom oznaczający, że wierzchołek należy do V_A (`e` to konkretny atom oznaczający, że wierzchołek należy do V_E), `v1,...,vn` to dowolne atomy będące unikalnymi identyfikatorami wierzchołków $v_1, \ldots, v_n$, przy czym dla każdego $i = 1, \ldots, n$ mamy $\langle v, v_i\rangle \in R$.

Reprezentacja AE-grafu $\langle V_A, V_E, R\rangle$ to lista reprezentacji wierzchołków z $V_A \cup V_E$. Reprezentacja grafu $\langle V, S\rangle$ to lista reprezentacji wierzchołków z V.

Na przykład dla AE-grafu:

- $V_A = \{v_0, v_2, v_3\}$,
- $V_E = \{v_1\}$,

- $R = \{\langle v_0, v_1\rangle, \langle v_0, v_3\rangle, \langle v_1, v_2\rangle, \langle v_1, v_3\rangle\}$.

Reprezentacja wierzchołka `v0` to lista `[v0, a, v1, v3]`, zaś reprezentacja całego AE-grafu to lista

```
[ [v0, a, v1, v3],
  [v1, e, v2, v3],
  [v2, a],
  [v3, a] ]
```

Dla tego AE-grafu mamy dwa wybory z niego. Jednym z nich jest graf

- $V = \{0, 1, 2, 3\}$,
- $S = \{\langle 0, 1\rangle, \langle 0, 3\rangle, \langle 1, 2\rangle\}$.

Jego reprezentacja to lista

```
[ [v0, a, v1, v3],
  [v1, e, v2],
  [v2, a],
  [v3, a] ]
```

Lista przejścia DFS dla tego grafu to `[v0,v1,v2,v3]`. Ta lista jest też listą alternującego przejścia DFS przez określony powyżej AE-graf $\langle V_A, V_E, R\rangle$.

Warto zwrócić uwagę, że określona tutaj metoda reprezentacji nie jest jednoznaczna, w szczególności podane powyżej reprezentacje nie są jedynymi możliwymi reprezentacjami wspomnianych grafów.

2 Zadanie

Zdefiniować następujące predykaty:

- `jestWyborem(+AEgraf, -Graf)` – odnosi sukces, gdy `Graf` jest wyborem z `AEgraf`. Gdy dla `AEgraf` istnieje wiele wyborów, predykat powinien odnosić wielokrotnie sukces, przynajmniej raz dla każdego wyboru.

- `jestDFS(+Graf, -Lista)` – odnosi sukces, gdy `Lista` jest listą identyfikatorów wierzchołków kolejno odwiedzanych przez algorytm przechodzenia grafu `Graf` w głąb przy przejściu startującym z pierwszego wierzchołka tego grafu.

- `jestADFS(+AEgraf, -Lista)` – odnosi sukces, gdy `Lista` jest listą identyfikatorów wierzchołków kolejno odwiedzanych przez algorytm przechodzenia w głąb przy przejściu przez pewien graf będący wyborem z `AEgraf`. W definicji tego predykatu należy jawnie zbudować reprezentację pewnego wyboru AE-grafu `AEgraf`. Można tego dokonać za pomocą predykatu `jestWyborem`.

- `jestADFS1(+AEgraf, -Lista)` – odnosi sukces, gdy `Lista` jest listą identyfikatorów wierzchołków kolejno odwiedzanych przez algorytm przechodzenia w głąb przy przejściu przez pewien graf będący wyborem z `AEgraf`. W trakcie obliczania tego predykatu nie może jawnie być budowana reprezentacja wyboru AE-grafu `AEgraf`.

2.1 Oddawanie i ocena rozwiązań

Rozwiązania muszą być całkowicie samodzielne. W szczególności wszelkie zapożyczenia z internetu oraz prace zbiorowe są niedozwolone.

Rozwiązanie powinno składać się z jednego pliku o nazwie

- <identyfikator_studenta>.pl, np. `ab123456.pl`,

który należy przesłać przez Moodle. Pierwszy wiersz pliku powinien zawierać komentarz z nazwiskiem autora. W przypadku oddania rozwiązania cząstkowego należy napisać w komentarzu, co zostało zrobione i co nie zostało. Plik nie powinien importować żadnych innych plików.

W rozwiązaniu wolno korzystać jedynie:

- z predykatów/konstrukcji przedstawionych na wykładzie,
- z tzw. wbudowanych predykatów (np. member/2, append/3 itp.),
- ze standardowej biblioteki SICStus Prologu o nazwie `lists` (ładowanie: `:- use_module(library(lists))`).

Nie wolno korzystać:

- z żadnych innych bibliotek,
- z wbudowanych predykatów nieprzedstawionych na wykładzie.

2.2 Kryteria oceny

Oceny będą wystawiane według cennika:

1. za poprawną implementację predykatów `jestWyborem` i `jestDFS` po jednym punkcie (łącznie do 2 punktów),
2. za poprawną implementację predykatów `jestADFS` i `jestADFS1` po dwa punkty (łącznie do 4 punktów),
3. za realizację zaimplementowanych predykatów pozwalającą na podawanie zapytań z nieustalonymi oboma parametrami predykatów i uzyskiwanie wszystkich odpowiedzi – jeden punkt; maksymalną ilość punktów można dostać tutaj także za realizację, która odnosi sukces dla wszystkich grafów lub list, ale niekoniecznie dla ich wszystkich reprezentacji,
4. za czytelność i klarowną dokumentację zaimplementowanych predykatów w komentarzach – jeden punkt.

Rozwiązania mogą korzystać z założenia, że argumenty wejściowe predykatów są prawidłowo zbudowane.