Fórum studentů MFF UK

Fórum pro všechny studenty matematicko-fyzikální fakulty UK, informatiky, fyziky i matematiky

Přejít na obsah

- Obsah fóra ‹ Informatika LS ‹ Výuka LS 2. ročník ‹ PRG005 Neprocedurální programování
- Změnit velikost textu
- Napsat e-mail
- Verze pro tisk

- FAQ
- Registrovat
- Přihlásit se

Zkouška 24.6.2019 (Dvořák, Hric)

Odeslat odpověď

- Ohlásit tento příspěvek
- Odpovědět s citací

Zkouška 24.6.2019 (Dvořák, Hric)

od **awk** » 24. 6. 2019 13:44

1. **Prolog: Generování výrokových formulí** (5 bodů)
   Formule výrokového počtu jsou sestavené z (výrokových) proměnných ve funktoru var/1 a logických spojek negace, konjunkce a disjunkce (bez konstant). Dále máte dány v argumentech predikátu gen/3 číslo k pro velikost formule a seznam jmen proměnných. Generujte backtrackingem všechny logické formule (každou jednou), které obsahují proměnné ze seznamu a ve kterých je počet spojek a výskytů proměnných dohromady právě k.

   Definujte predikát gen(+K, +Jmena, -Fle). Na pořadí generovaných formulí nezáleží, ale měli byste vygenerovat každou právě jednou. K řešení není potřeba predikát =../2 (univ).

   Příklad:

   ?- gen(4,[p],F).

   F = not(not(not(var(p))));
   F = not(and(var(p),var(p)));
   F = not(or(var(p),var(p)));
   F = and(not(var(p)),var(p));

F = and(var(p),not(var(p)));
F = or(not(var(p)),var(p));
F = or(var(p),not(var(p)));
false.

2. **Prolog: Koncepty** (5 bodů)
Jeden objekt je zadán uspořádaným seznamem dvojic klíč-hodnota. Na vstupu máte seznam objektů. Napište proceduru koncept/2, která vyrobí nejmenší koncept zahrnující všechny vstupní objekty. Koncept je seznam dvojic klíč-seznam_hodnot. Koncept zahrnuje objekt, pokud koncept má všechny klíče objektu a v seznamu hodnot příslušného klíče u konceptu je obsažena hodnota klíče u objektu. Pokud objekt nějaký klíč konceptu nemá, bude v seznamu hodnot konceptu hodnota nedef.

Příklad:

?- koncept([[barva-modra, motor-diesel, pocet_kol-6], [barva-bila, motor-plyn, pocet_mist-40], [motor-elektro, pocet_mist-5]], Koncept).

Koncept = [barva-[modra,bila,nedef], motor-[diesel,plyn,elektro], pocet_kol-[6,nedef], pocet_mist-[40,5,nedef]]

3. **Haskell: Kumulativní součty** (5 bodů)
Je dána číselná matice A. Definujte funkci

kumulace :: Num a => [[a]] -> [[a]]

která z matice A vyrobí matici B stejných rozměrů (viz příklad níže)

každý prvek na souřadnicích (i,j) bude roven součtu všech hodnot
v submatici s levým horním rohem (0,0)
a pravým dolním rohem (i,j)

Poznámka: Snažte se vyhnout opakování stejných výpočtů.

Příklad:

> kumulace[[1,1,1],[1,2,1],[0,1,0],[1,1,-4]]
[[1,2,3],[2,5,7],[2,6,8],[3,8,6]]

4. **Haskell: Doplnění hypergrafu** (5 bodů)
Hypergraf je zadán množinou vrcholů a množinou hyperhran, což jsou alespoň dvouprvkové podmnožiny množiny vrcholů. Naší cílem je definovat funkci doplnění, která doplní do hypergrafu H všechny dvouprvkové (hyper)hrany pro ty dvojice vrcholů, které nejsou společně obsaženy v žádné hyperhraně vstupního hypergrafu H. Funkce tedy např. z hypergrafu
   - s vrcholy {1,2,3,4,5} a hyperhranani {1,3,5} a {2,3,4}
   - vytvoří hypergraf se stejnými vrcholy a hyperhranami {1,3,5},{2,3,4},{1,2},{1,4},{5,2} a {5,4}

(a) Definujte datový typ pro reprezentaci hypergrafu. Pokuste se o co nejobecnější definici (vrcholy mohou být reprezentovány nejen čísly, ale i znaky, řetězci apod.)

(b) Specifikujte typovou signaturu funkce

doplneni ::

(c) Funkci definujte.

Naposledy upravil awk dne 24. 6. 2019 15:08, celkově upraveno 2

awk

Matfyz(ák|ačka) level II

Příspěvky: 50
Registrován: 21. 5. 2018 17:54
Typ studia: Informatika Bc.

Nahoru

- Ohlásit tento příspěvek
- Odpovědět s citací

Re: Zkouška 24.6.2019 (Dvořák, Hric)

od **matfiz** » 24. 6. 2019 15:00

Velký příklad:

Klastrování pro detektor (Prolog/Haskell)

Data z detektoru jsou dlouhý seznam událostí z jednotlivých senzorů, které jsou uspořádány ve čtvercové mřížce. Jedna částice zasáhne i víc senzorů. Každý zasažený senzor pošle svojí polohu (v mřížce), energii částice a čas příletu. Máte spojit do klastrů informace o jedné částici. Klastr je co největší, ale spojitý (4-souvislost, po hranách mřížky) a všechny časy příletů se liší nejvýš o Delta. Problém je, že data přicházejí zpožděny až o Z, Z > Delta, ale víte, že senzor nepošle dvě události během doby Z. Několik částic může přiletět současně a následně data z klastrů jsou na vstupu promíchána.

Napište funkci/proceduru klastry, která dostane na vstup Delta, Z a seznam událostí jako čtveřic (i,j, energie, čas). Na výstup posíláte seznam klastrů, kde každý klastr je seznam událostí.

matfiz

Nahoru

- Ohlásit tento příspěvek
- Odpovědět s citací

Re: Zkouška 24.6.2019 (Dvořák, Hric)

od **awk** » 25. 6. 2019 11:42

Skvělý, díky za big one.

Dnes jsem byl na ústní zkoušce u Dvořáka, byl moc hodný .

Letmo jsme projeli velký příklad — zajímala ho nějaká myšlenka odevzdaného kódu. Pak jsme se hlavně věnovali malým příkladům. Probíhalo to formou příjemné diskuse ohledně odevzdaného řešení. Diskutovali se možná zlepšení, chyby, ale i dobře vyřešené příklady, kde poukazoval na kusy kódu, které se mu líbili.

Celkově mi připadá důležité se na odevzdaná řešení před ústní zkouškou podívat a zjistit, co opravdu funguje a co ne. Pokud to nefunguje, tak si najít chybu, případně zjistit, kolik úprav je třeba udělat, aby se dané řešení

zfunkčnilo. Může vám to hodně pomoci.

Počítejte s tím, že se vás může zeptat i na nějakou doplňující otázku, mě se zeptal na rozdíl mezi *type*, *newtype* a *data* v Haskellu. Vyplatí se umět i různé seznamoviny (hlavně věci kolem rozdílových seznamů). Pokud byste chtěli lepší známku, než na jakou dle vašeho výkonu v písemné části máte, tak můžete počítat s tím, že dostanete něco opravit/naprogramovat na papír.

awk

Matfyz(ák|ačka) level II

Příspěvky: 50
Registrován: 21. 5. 2018 17:54
Typ studia: Informatika Bc.

Nahoru

- Ohlásit tento příspěvek
- Odpovědět s citací

Re: Zkouška 24.6.2019 (Dvořák, Hric)

od **awk** » 26. 6. 2019 06:47

Přidávám ještě moje řešení malých příkladů.

1. **Prolog: Generování výrokových formulí** (5 bodů)

Kód: Vybrat vše

```
gen(1, Jmena, var(J)) :- !, member(J, Jmena).
gen(2, Jmena, not(P)) :- !, gen(1, Jmena, P).
gen(N, Jmena, V) :-
    N > 2,
    B is N-2,
    C is N-1,
    between(1,B,X),
    D is C - X,
    gen(X, Jmena, P),
    gen(D, Jmena, F),
    ( V = and(F,P) ; V = or(F,P) ).

gen(N, Jmena, not(E)) :-
    N > 2,
    A is N-1,
    gen(A, Jmena, E).
```

2. **Prolog: Koncepty** (5 bodů)

Kód: Vybrat vše

```
vstup(V) :-
    V = [
            [barva-modra,motor-diesel, pocet_kol-6],
```

```
            [motor-plyn,pocet_mist-40, barva-modra],
            [motor-elektro,pocet_mist-5]
        ].


foldr(_, [], Acc, Acc) :- !.
foldr(Binary, [X | Xs], Acc, Y) :-
    foldr(Binary, Xs, Acc, Z),
    call(Binary, X, Z, Y).

foldl(_, [], Y, Y) :- !.
foldl(Binary, [X | Xs], Acc, Y) :-
    call(Binary, X, Acc, NewAcc),
    foldl(Binary, Xs, NewAcc, Y).


filter(_, [], []) :- !.
filter(Pred, [X | Xs], [X | Ys]) :-
    call(Pred, X),
    !,
    filter(Pred, Xs, Ys).

filter(Pred, [_ | Xs], Ys) :- filter(Pred, Xs, Ys).

unique([],[]).
unique([X | Xs], [X | Ys]) :-
    filter(notUnifiable(X), Xs, Z),
    unique(Z, Ys).

notUnifiable(X, Y) :- X \= Y.

fst(H, Y) :- H=..[_, Y, _].

map(_, [], []) :- !.
map(Unary, [X | Xs], [Y | Ys]) :-
    call(Unary, X, Y),
    map(Unary, Xs, Ys).

koncept2nazvy(O, N) :- map(fst, O, N).

pridejNazvy(O, Acc, NewAcc) :-
    koncept2nazvy(O, N),
    append(Acc, N, NewAcc).

koncepty2nazvy(Os, Ns) :-
    foldl(pridejNazvy, Os, [], Nd),
    unique(Nd, Ns).

vyndej(Name, A-_, Acc, Acc) :- Name \= A, !.
vyndej(Name, Name-B, Acc, [B | Acc]).

getAttributeValue(O, Name, Value) :-
    foldl(vyndej(Name),O,[], Values),
    length(Values, L),
    (L == 1 ->
        Values = [Value] ;
    (L == 0 ->
        Value = nedef ;
```

```
    %else
        fail
    )
    ).


p(Name, O, Acc, [Value|Acc]) :- getAttributeValue(O, Name, Value), !.
p(_, _, Acc, Acc).

getAttributeValues(Os, Name, Values) :- foldr(p(Name),Os, [], Values).

dvojce(Objekty, Nazev, Nazev-Hodnoty) :-
    getAttributeValues(Objekty, Nazev, Vals),
    unique(Vals, Hodnoty).


koncept(V, Y) :-
    koncepty2nazvy(V, Ns),
    map(dvojce(V),Ns,Y).
```

3. **Haskell: Kumulativní součty** (5 bodů)

Kód: Vybrat vše

```
safehead :: [[a]] -> [a]
safehead xs | null xs   = []
            | otherwise = head xs

kumulace :: Num a => [[a]] -> [[a]]
kumulace m = reverse $ foldl (\km r -> kumulaceRadku (safehead km) r : km) [] m

kumulaceRadku :: Num a => [a] -> [a] -> [a]
kumulaceRadku [] r = scanl1 (+) r
kumulaceRadku kr r = foldr (\(a,b,c,d) acc -> a+b+c-d : acc) [] xs
    where xs = zip4 kr r

zip4 :: Num a => [a] -> [a] -> [(a,a,a,a)]
zip4 kr r = zip4' kr r (0:kr) 0
    where
        zip4' :: Num a => [a] -> [a] -> [a] -> a -> [(a,a,a,a)]
        zip4' _ [] _ _ = []
        zip4' (a : kr) (b : r) (c : kr1) d = (b,d,a,c) : zip4' kr r kr1 (b+d+a-c)
```

4. **Haskell: Doplnění hypergrafu** (5 bodů)

Kód: Vybrat vše

```
data Hypergraf a = Hypergraf [a] [[a]] deriving Show

testHypergraf = Hypergraf [1,2,3,4,5]  [[1,3,5], [2,3,4]]

prvek :: Eq a => [a] -> [[a]] -> Bool
prvek _ [] = False
prvek x (y : ys) = jePrvek || jeObracenePrvek || prvek x ys
        where
            jePrvek = x == y
            jeObracenePrvek = reverse x == y
```

```
difference :: Eq a => [[a]] -> [[a]] -> [[a]]
difference xs ys = foldr (\x a -> if prvek x a then a
                                  else if not $ prvek x ys then (x : a)
                                  else a
                         ) [] xs

doplneni :: Eq a => Hypergraf a -> Hypergraf a
doplneni (Hypergraf xs ys) = Hypergraf xs (ys ++ hranyPridat)
        where
             vsechnyMozneHrany = kombinace 2 xs
             hranyUvnitr = foldr (\hhrana ak -> ak ++ kombinace 2 hhrana) [] ys
             hranyPridat = difference vsechnyMozneHrany hranyUvnitr


kombinace :: (Eq a, Integral b) => b -> [a] -> [[a]]
kombinace 0 _ = [[]]
kombinace _ [] = []
kombinace n (x : xs) | n > 0 = kombinace n xs ++ map (x:) (kombinace (n-1) xs)
```

awk

Matfyz(ák|ačka) level II

Příspěvky: 50
Registrován: 21. 5. 2018 17:54
Typ studia: Informatika Bc.

Nahoru

Zobrazit příspěvky za předchozí: Všechny příspěvky Seřadit podle Čas odeslání Vzestupně Přejít

Odeslat odpověď

Příspěvků: 4 • Stránka **1** z **1**

Zpět na PRG005 Neprocedurální programování

Přejít na: PRG005 Neprocedurální programování Přejít

Kdo je online

Uživatelé procházející toto fórum: Žádní registrovaní uživatelé a 1 návštěvník

- Obsah fóra
- Tým • Smazat všechny cookies z fóra • Všechny časy jsou v UTC + 1 hodina