IB015 Zápočet, skupina B 60 minut **Datum: 18. 12. 2018**

Jméno: **UČO:** **Souřadnice:**

0007 *list* | *učo* *body*

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0123456789

Pro udělení zápočtu je třeba vyřešit správně všech pět příkladů v této písemce. Jedna drobná chyba bude tolerována.

1. [Haskell] Naprogramujte funkci `myDropWhile :: (a → Bool) → [a] → [a]`, která se chová stejně jako její obdoba ze standardní knihovny `dropWhile`. Tato funkce vrátí zadaný seznam, ze kterého je vynechán nejdelší možný prefix, pro jehož prvky platí, že funkce zadaná jako první argument na nich vrátí hodnotu `True`.

Můžete využít libovolné konstrukce jazyka Haskell, nesmíte však použít knihovní funkci `dropWhile`.

Příklady vyhodnocení:

```
myDropWhile odd [1, 3, 2, 3] ⇝* [2, 3]
myDropWhile (\_ → False) [1, 2, 3] ⇝* [1, 2, 3]
myDropWhile even [] ⇝* []
```

```
myDropWhile::(a -> Bool) -> [a] -> [a]
myDropWhile _ []     = []
myDropWhile f (x:xs) = if f x then myDropWhile f xs
                              else x:xs
```

2. [Haskell] Pro následující výraz programovacího jazyka Haskell uveďte, čím nahradit text `???` tak, aby platilo uvedené vyhodnocení.

```
filter ??? [1..10] ⇝* [5,6,7,8,9,10]
```

```
(>4)
```

Oblast strojově snímaných informací, nezasahujte. ***Druhá strana se neskenuje.***

IB015 Zápočet, skupina B 60 minut Datum: 18. 12. 2018

Jméno: UČO: Souřadnice:

0007 *list* 2 *učo* *body*

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0123456789

3. [Haskell] Mějme dán datový typ `EMail`, jehož hodnoty slouží k modelování emailového diskuzního vlákna (konverzace), kdy na nějaký původní email (hodnota typu `String`) se následně kumulují odpovědi (hodnoty typu `String`), a to tak, že celá historie komunikace je vždy zachována.

```
data EMail = Text String
           | Reply String EMail
```

a) Uveďte libovolnou platnou hodnotu typu `EMail`, ve které je použit hodnotový konstruktor `Reply`.

b) Napište funkci `conLen :: EMail → Int`, která pro libovolnou hodnotu typu `EMail` spočítá délku konverzace (počet zpráv v konverzaci).

a)

```
Reply "reply" (Text "String")
```

b)

```
conLen:: EMail -> Int
conLen (Text _) = 1
conLen (Reply _ e) = 1 + conLen e
```

Oblast strojově snímaných informací, nezasahujte. ***Druhá strana se neskenuje.***

IB015 Zápočet, skupina B | 60 minut | Datum: 18. 12. 2018

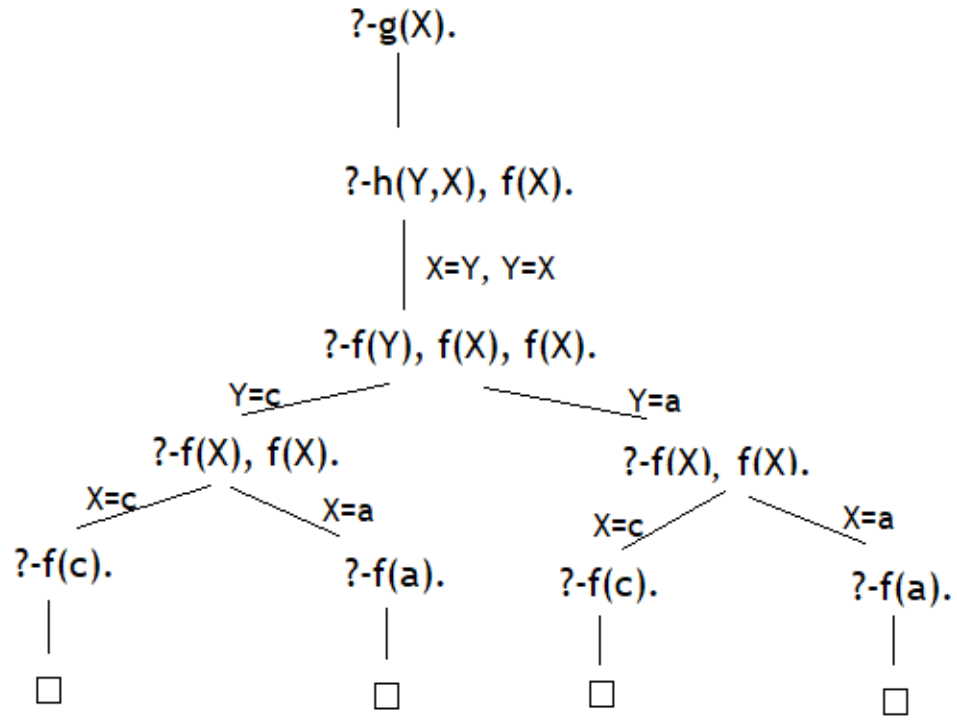
Jméno: | UČO: | Souřadnice:

0007 list 3 učo body

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0 1 2 3 4 5 6 7 8 9

4. [Prolog] Vypište ve správném pořadí všechny možné odpovědi (uživatel vkládá středník), které interpret Prologu vrátí na dotaz `?- g(X).` za předpokladu následující databáze pravidel a faktů.

```
f(c).
f(a).
g(X) :- h(Y,X), f(X).
h(X,Y) :- f(X), f(Y).
```



X=c, X=a, X=c, X=a.

Oblast strojově snímaných informací, nezasahujte. ***Druhá strana se neskenuje.***

IB015 Zápočet, skupina B | 60 minut | Datum: 18. 12. 2018

Jméno: | UČO: | Souřadnice:

0007 *list* 4 *učo* *body*

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0123456789

5. [Prolog] Do programovacího systému Prolog převeďte níže uvedenou funkci `foo` programovacího jazyka Haskell, tj. definujte predikát `myFoo(+XS,?YS)`, který uspěje právě pro takové dvojice seznamů čísel, pro které platí, že aplikace `foo` na první seznam vrátí druhý seznam. Můžete předpokládat, že všechny seznamy jsou neprázdné.

Funkce v Haskellu:

```
foo :: [[Integer]] → [Integer]
foo xs = map head xs
```

```
myFoo ([], YS):- YS = [].
myFoo([[x|_]|XS], YS):- myFoo(XS, Y), YS = [X|Y].
```