

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 písence. Jedna drobná chyba bude tolerována.

1. [Haskell] Naprogramujte funkci `myTakeWhile :: (a → Bool) → [a] → [a]`, která se chová stejně jako její obdoba ze standardní knihovny `takeWhile`. Tato funkce vrátí nejdelší možný prefix seznamu zadaného jako druhý argument, pro jehož všechny prvky platí, že funkce zadaná jako první argument vrátí hodnotu `True`.

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

Příklady vyhodnocení:

```
myTakeWhile odd [1, 3, 2, 3] ~* [1, 3]
```

```
myTakeWhile (\_ → False) [1, 2, 3] ~* []
```

```
myTakeWhile even [] ~* []
```

$$\text{myTakeWhile} :: (a \rightarrow \text{Bool}) \rightarrow [a] \rightarrow [a]$$

$$\text{myTakeWhile} _ [] = []$$

$$\text{myTakeWhile } f (x:xs) = \text{if } (f\ x) \text{ then } x:\text{myTakeWhile } f\ xs \text{ else } []$$

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

```
map ??? [1,2,3,4] ~* [1,4,9,16]
```

$$(\wedge 2)$$

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 `Message`, jehož hodnoty slouží k modelování situace, kdy je nějaká původní textová zpráva (hodnota typu `String`) libovolněkrát přeposlána s dalším komentářem (opět hodnota typu `String`).

```
data Message = Original String
              | Forwarded String Message
```

a) Uvedte libovolnou platnou hodnotu typu `Message`, ve které je použit hodnotový konstruktor `Forwarded`.

b) Napište funkci `getOriginal :: Message → String`, která pro libovolnou hodnotu typu `Message` vrátí původní zprávu.

a) ~~Forwarded "ahoj" (Original "světe")~~
a) Forwarded "ahoj" (Original "světe")

b) `getOriginal :: Message → String`
`getOriginal (Original s) → s`
`getOriginal (Forwarded o) → getOriginal o`

Jméno:

UČO:

Souřadnice:

0007

list

3

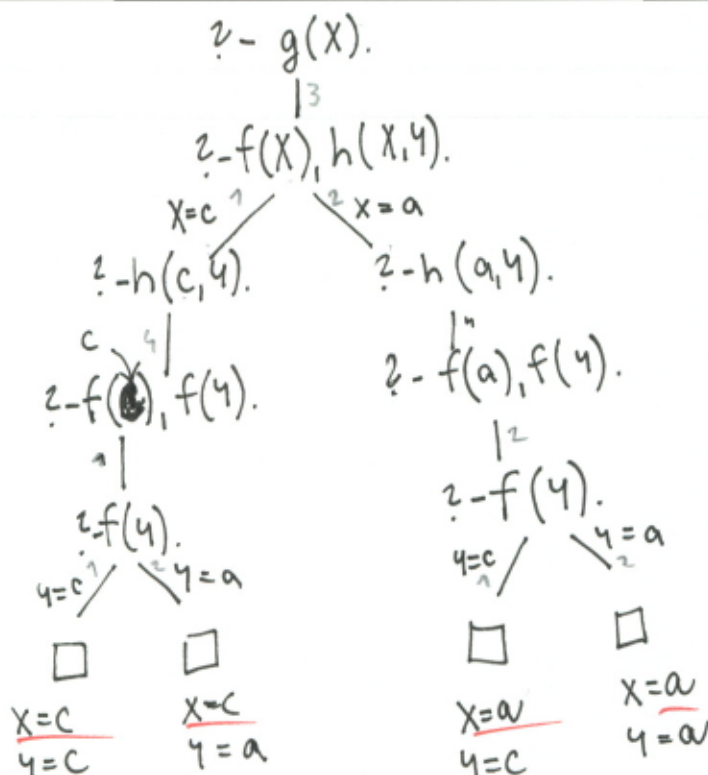
učo

body

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

0123456789

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.

1 $f(c)$.2 $f(a)$.3 $g(X) :- f(X), h(X,Y)$.4 $h(X,Y) :- f(X), f(Y)$.

Handwritten red text: *Kodování*

Jméno:

UČO:

Souřadnice:

0007

list

4

učo

body

0

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 převed'te 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.

Funkce v Haskellu:

```
foo :: [Integer] → [Integer]
foo xs = filter (> 3) xs
```

[1,5,8]

[5,8]

[5,8]

myFoo(+XS,?YS)

myFoo([], []).

myFoo([x|XS], [x|YS]) :- x > 3, myFoo(XS, YS).

myFoo([-|XS], YS) :- myFoo(XS, YS).

✗ x < 3