



list

učo

body

Oblast strojově snímaných informací. Svě 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 `myAll :: (a → Bool) → [a] → Bool`, která se chová stejně jako její obdoba ze standardní knihovny `all`. Tato funkce vrátí hodnotu `True`, pokud všechny prvky seznamu zadaného jako druhý argument splňují podmínku zadanou jako první argument.

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

Příklady vyhodnocení:

```
myAll odd [1,3,2,3] ~* False
```

```
myAll (\_ → True) [1,2,3] ~* True
```

```
myAll even [] ~* True
```

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

$$\text{myAll } f [] = \text{True}$$

$$\text{myAll } f (x:xs) = \text{if } f x \text{ then } \text{myAll } f xs \text{ else False}$$

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

```
??? odd [1..10] ~* [1,3,5,7,9]
```

filter

0007

list

2

učo

body

Oblast strojově snímaných informací. Svě 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) Uveďte dvě hodnoty typu `Message`, které jsou strukturálně odlišné (tj. způsob použití hodnotových konstruktorů je různý).

b) Napište funkci `getNumForwards :: Message → Int`, která pro libovolnou hodnotu typu `Message` vrátí počet, kolikrát byla původní zpráva přeposlána.

b) $\text{getNumForwards (Original msg)} = 0$
 $\text{getNumForwards (Forwarded msg next)} = 1 + (\text{getNumForwards next})$

2) $(\text{Original "hello"})$

$(\text{Forwarded "comment" (Original "text")})$

0007

list

3

učo

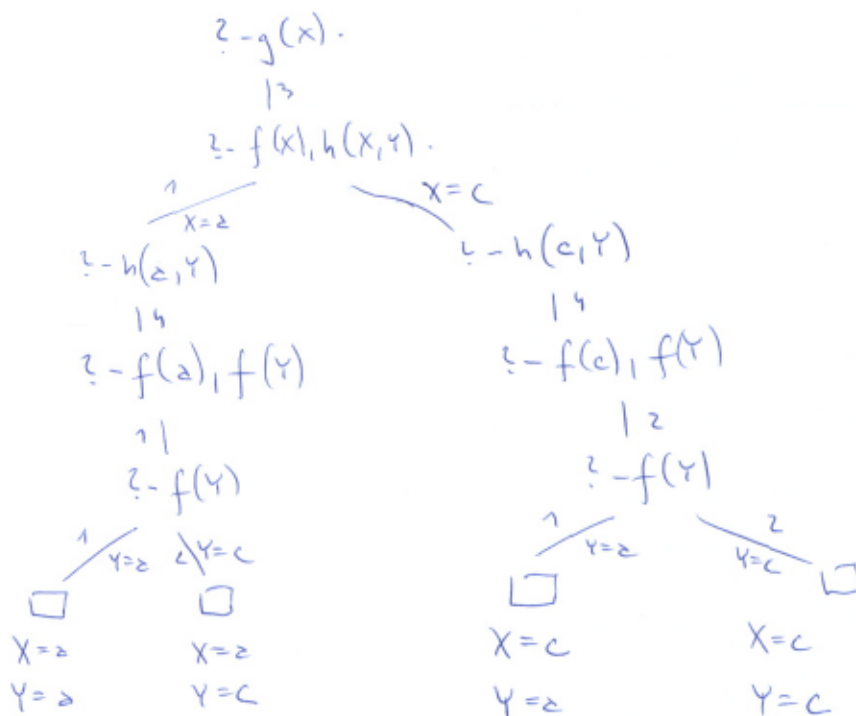
body

Oblast strojově snímaných informací. Svě 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.

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



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

0007

list

4

učo

body

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

0123456789

5. [Prolog] Do programovacího systému Prolog převedte 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 tail xs
```

$\{[1,2,3], [2,2,2], [3,6,7,8], [1], [1,1]\}$

$\{[2,3], [2,2], [6,7,8], [], [1]\}$

$\text{myFoo}([X|E], [F|YS]) :- \text{myFoo}(XS, YS).$
 $\text{myFoo}([], []).$