Programowanie 2009 Programming 2009

Lista zadań nr 11 Problem set no. 11

Na zajęcia 19–20 maja 2009 Due May 20, 2009

Grupy zasadnicze Basic groups

Oto fragment Haskella: Here is a fragment of Haskell:

$$
\begin{array}{rcll}
map & :: & (a \to b) \to [a] \to [b] & \\
map\ \_\ [] & = & [] & (1) \\
map\ f\ (x : xs) & = & f\ x : map\ f\ xs & (2) \\
 & & & \\
filter & :: & (a \to Bool) \to [a] \to [a] & \\
filter\ \_\ [] & = & [] & (3) \\
filter\ p\ (x : xs) & & & \\
\quad \mid p\ x & = & x : filter\ p\ xs & (4) \\
\quad \mid otherwise & = & filter\ p\ xs & (5) \\
 & & & \\
(+\!\!+) & :: & [a] \to [a] \to [a] & \\
[] +\!\!+ ys & = & ys & (6) \\
(x : xs) +\!\!+ ys & = & x : (xs +\!\!+ ys) & (7) \\
 & & & \\
concat & :: & [[a]] \to [a] & \\
concat\ [] & = & [] & (8) \\
concat\ (xs : xss) & = & xs +\!\!+ concat\ xss & (9) \\
 & & & \\
(.) & :: & (b \to c) \to (a \to b) \to (a \to c) & \\
(f\ .\ g)\ x & = & f\ (g\ x) & (10) \\
 & & & \\
id & :: & a \to a & \\
id\ x & = & x & (11) \\
 & & & \\
foldr & :: & (a \to b \to b) \to b \to [a] \to b & \\
foldr\ \_\ c\ [] & = & c & (12) \\
foldr\ (\oplus)\ c\ (x : xs) & = & x \oplus foldr\ (\oplus)\ c\ xs & (13) \\
 & & & \\
foldl & :: & (b \to a \to b) \to b \to [a] \to b & \\
foldl\ \_\ c\ [] & = & c & (14) \\
foldl\ (\otimes)\ c\ (x : xs) & = & foldl\ (\otimes)\ (c \otimes x)\ xs & (15) \\
 & & & \\
flip & :: & (a \to b \to c) \to (b \to a \to c) & \\
flip\ f\ x\ y & = & f\ y\ x & (16) \\
 & & & \\
reverse & :: & [a] \to [a] & \\
reverse\ [] & = & [] & (17) \\
reverse\ (x : xs) & = & reverse\ xs +\!\!+ [x] & (18)
\end{array}
$$

$$
\begin{aligned}
rev &:: [a] \to [a] \\
rev &= aux\ []\ \textbf{where} && (19)\\
&\quad aux\ xs\ [] = xs && (20)\\
&\quad aux\ xs\ (y : ys) = aux\ (y : xs)\ ys && (21)
\end{aligned}
$$

Zbadaj, które z poniższych faktów zachodzą dla dowolnych list, a które tylko dla skończonych. Udowodnij odpowiednio sformułowane fakty i podaj kontrprzykłady, gdy równości nie zachodzą dla list nieskończonych.

Investigate which facts below hold for arbitrary lists, and which only for finite ones. Prove properly stated facts and give counterexamples in cases where equalities do not hold for infinite lists.

Zadanie/Problem 1 (1 p). $map\ (f\ .\ g) = map\ f\ .\ map\ g$.

Zadanie/Problem 2 (1 p). $map\ f\ (xs +\!\!+ ys) = map\ f\ xs +\!\!+ map\ f\ ys$.

Zadanie/Problem 3 (1 p). $f\ .\ g = id \implies (map\ f)\ .\ (map\ g) = id$.

Zadanie/Problem 4 (1 p). $map\ f\ .\ concat = concat\ .\ map\ (map\ f)$.

Zadanie/Problem 5 (1 p). $filter\ p\ (xs +\!\!+ ys) = filter\ p\ xs +\!\!+ filter\ p\ ys$.

Zadanie/Problem 6 (1 p). $filter\ p\ .\ concat = concat\ .\ map\ (filter\ f)$.

Zadanie/Problem 7 (1 p). $reverse = rev$.

Zadanie/Problem 8 (1 p). $reverse = foldl\ (flip\ (:))\ []$.

Zadanie 9 (1 pkt). Wiemy, że dla list skończonych

Problem 9 (1 p). We know that for finite lists

$$reverse\,(reverse\,x) \quad = \quad x.$$

Spróbuj udowodnić tę równość dla wszystkich list (w tym nieskończonych) korzystając z zasady indukcji omówionej na wykładzie i pokaż, gdzie dowód się zacina.

Try to prove this equality for all lists (including infinite ones) using the induction principle described during the lecture, and show where the proof breaks.

Grupy rozszerzone

Extended groups

Zadanie 1 (1 pkt). Niech $(\oplus) :: a \to b \to b$, $(\otimes) :: b \to a \to b$ i $a :: b$ spełniają następujące równości:

Problem 1 (1 p). Let $(\oplus) :: a \to b \to b$, $(\otimes) :: b \to a \to b$ and $a :: b$ satisfy the following equalities:

$$
\begin{aligned}
x \oplus (y \otimes z) &= (x \oplus y) \otimes z, && (22)\\
x \oplus a &= a \otimes x, && (23)
\end{aligned}
$$

dla dowolnych $x, z :: a$ i $y :: b$. Udowodnij, że dla dowolnej *skończonej* listy $xs :: [a]$ zachodzi następująca równość:

for any $x, z :: a$ and $y :: b$. Prove that for any *finite* list $xs :: [a]$ the following equality holds:

$$foldr\ (\oplus)\ a\ xs \quad = \quad foldl\ (\otimes)\ a\ xs.$$

Jest to tzw. *drugie prawo dualności*. Wywnioskuj zeń tzw. *pierwsze prawo dualności*: jeśli $\langle (\oplus) :: a \to a \to a, a :: a \rangle$ tworzą monoid, to dla dowolnej skończonej listy $xs :: [a]$ zachodzi równość:

This is so called *the second duality law*. Infer from it so called *the first duality law*: if $\langle (\oplus) :: a \to a \to a, a :: a \rangle$ form a monoid, then for any finite list $xs :: [a]$ the following equality holds:

$$foldr\ (\oplus)\ a\ xs \quad = \quad foldl\ (\oplus)\ a\ xs.$$

Wskaż przykłady, że powyższe prawa nie zawsze są słuszne dla list nieskończonych.

Give examples showing that these laws may not be satisfied for inifite lists.

Zadanie 2 (1 pkt). Udowodnij, że dla dowolnej *skończonej* listy $xs :: [a]$ zachodzi równość

Problem 2 (1 p). Prove that for any *finite* list $xs :: [a]$ the following equality holds:

$$\mathit{foldr}\ (\oplus)\ a\ xs \quad = \quad \mathit{foldl}\ (\mathit{flip}\ (\oplus))\ a\ (\mathit{reverse}\ xs).$$

Jest to tzw. *trzecie prawo dualności*. Wskaż przykład dowodzący, że może ono nie być spełnione dla listy nieskończonej.

It is called *the third duality law*. Give an example showing that it may not be satisfied for inifite lists.

Zadanie 3 (1 pkt). Udowodnij, że

Problem 3 (1 p). Prove that

$$\mathit{foldl}\ (\otimes)\ a\ (xs +\!\!+\ ys) \quad = \quad \mathit{foldl}\ (\otimes)\ (\mathit{foldl}\ (\otimes)\ a\ xs)\ ys.$$

Udowodnij następnie, że jeśli listy xs i ys są skończone, to

Next prove that if lists xs and ys are finite, then

$$\mathit{reverse}\ (xs +\!\!+\ ys) \quad = \quad \mathit{reverse}\ ys +\!\!+\ \mathit{reverse}\ xs.$$

Korzystając z tych równości i trzeciego prawa dualności wyprowadź podobną równość dla *foldr*.

Using these equalities and the third duality law derive a similar equality for *foldr*.

Zadanie 4 (1 pkt). Udowodnij, że

Problem 4 (1 p). Prove that

$$\mathit{map}\ f\ .\ \mathit{reverse} = \mathit{reverse}\ .\ \mathit{map}\ f.$$

Korzystając z tej równości i praw dualności udowodnij, że jeżeli $x \otimes y = x \oplus f\ y$, to

Using this equality and the duality laws prove, that if $x \otimes y = x \oplus f\ y$, then

$$\mathit{foldl}\ (\oplus)\ a\ .\ \mathit{map}\ f = \mathit{foldl}\ (\otimes)\ a.$$

Zadanie 5 (1 pkt). Niech $(\oplus) :: b \to b \to b$. Funkcja $h :: [a] \to b$ jest $\oplus$-*homomorficzna*, jeżeli dla wszelkich list $x, y :: [a]$ jest

Problem 5 (1 p). Let $(\oplus) :: b \to b \to b$. A function $h :: [a] \to b$ is $\oplus$-*homomorphic*, if for all lists $x, y :: [a]$ there is

$$h\ (x +\!\!+\ y) \quad = \quad h\ x \oplus h\ y.$$

Pokaż, że na zbiorze wartości funkcji $\oplus$-homomorficznej h operator $\oplus$ jest łączny, a lista pusta należy do dziedziny funkcji h, wtedy i tylko wtedy, gdy $h\ []$ jest elementem neutralnym względem $\oplus$.

Show that on the range of a $\oplus$-homomorphic function h the operator $\oplus$ is associative and the empty list belongs to the domain of the function h if and only if $h\ []$ is the unit of $\oplus$.

Dla dowolnego monoidu $\langle \oplus, e \rangle$ przez $\mathit{hom}\ (\oplus)\ f\ e$ będziemy oznaczać taką (unikatową) funkcję $\oplus$-homomorficzną h, dla której $h\ .\ (:[]) = f$. Udowodnij, że

For an arbitrary monoid $\langle \oplus, e \rangle$ we will write $\mathit{hom}\ (\oplus)\ f\ e$ for such a (unique) $\oplus$-homomorphic function h, for which $h\ .\ (:[]) = f$. Prove that

$$\mathit{map}\ f \quad = \quad \mathit{hom}\ (+\!\!+)\ ((:[])\ .\ f)\ [].$$

Zadanie 6 (1 pkt). Funkcja postaci $\mathit{hom}\ (\odot)\ \mathit{id}\ e$ nazywa się *redukcją*. Udowodnij, że funkcja jest homomorfizmem wtedy i tylko wtedy, gdy jest złożeniem redukcji z *map*:

Problem 6 (1 p). A function of the form $\mathit{hom}\ (\odot)\ \mathit{id}\ e$ is called a *reduction*. Prove that a function is a homomorphism if and only if it is the composition of a reduction and a map:

$$\mathit{hom}\ (\odot)\ f\ e \quad = \quad \mathit{hom}\ (\odot)\ \mathit{id}\ e\ .\ \mathit{map}\ f.$$

Zadanie 7 (1 pkt). Funkcja $h :: [a] \to b$ jest $\oplus$-*zlewna* (albo $\oplus$-*wprawna*), jeżeli dla dowolnych $a :: a$ i $y :: [a]$ jest

Problem 7 (1 p). A function $h :: [a] \to b$ is $\oplus$-*leftward* if for any $a :: a$ and $y :: [a]$ there is

$$h\,([a] +\!\!+\, y) \quad = \quad a \oplus h\; y.$$

Podobnie funkcja $h :: [a] \to b$ jest $\otimes$-*sprawna* (albo $\otimes$-*wlewna*), jeżeli dla dowolnych $x :: [a]$ i $a :: a$ jest

Similarly a function $h :: [a] \to b$ is $\otimes$-*rightward* if for any $x :: [a]$ and $a :: a$ there is

$$h\,(x +\!\!+\, [a]) \quad = \quad h\; x \otimes a.$$

Niech $\oplus$ i $\otimes$ będą dowolnymi operacjami (niekoniecznie łącznymi), a $e :: b$ — dowolnym elementem (niekoniecznie neutralnym względem $\oplus$ i $\otimes$). Udowodnij, że jeśli $h\,[] = e$ i h jest zlewna, to $h = \mathit{foldr}\,(\oplus)\; e$, a jeśli sprawna, to $h = \mathit{foldl}\,(\otimes)\; e$.

Let $\oplus$ and $\otimes$ be arbitrary (not necessarily associative) operations, and $e :: b$ — an arbitrary element (not necessarily neutral with respect to $\oplus$ or $\otimes$). Prove that if $h\,[] = e$ and h is leftward, then $h = \mathit{foldr}\;(\oplus)\; e$, and if rightward — then $h = \mathit{foldl}\,(\otimes)\; e$.

Udowodnij, że jeśli h jest homomorfizmem, to jest zarówno zlewny, jak i sprawny, tj. jeśli $\odot$ jest operatorem łącznym, to

Prove that if h is a homomorphism, than it is both leftward and rightward, *i.e.*, if $\odot$ is an associative operator, then

$$\begin{aligned} \mathit{hom}\,(\odot)\; f\; e \quad &= \quad \mathit{foldr}\,(\oplus)\; e, \qquad a \oplus s = f\; a \odot s, \\ &= \quad \mathit{foldl}\,(\otimes)\; e, \qquad r \otimes a = r \odot f\; a. \end{aligned}$$

Zadanie 8 (1 pkt). Niech

Problem 8 (1 p). Let

$$\begin{aligned} \mathit{sort} \quad &:: \quad \mathit{Ord}(a) \Rightarrow [a] \to [a] \\ \mathit{sort} \quad &= \quad \mathit{foldr}\; \mathit{ins}\; [] \\ \\ \mathit{ins} \quad &:: \quad \mathit{Ord}(a) \Rightarrow a \to [a] \to [a] \\ \mathit{ins}\; a\; (b : x) \quad &= \quad \begin{cases} a : b : x, & a \le b, \\ b : (\mathit{ins}\; a\; x), & \text{otherwise.} \end{cases} \end{aligned}$$

Funkcja ta jest oczywiście zlewna. Pokaż, że jest także sprawna.

This function is of course leftward. Show it is also rightward.

Zadanie 9 (1 pkt). Udowodnij, że dla dowolnych operacji $(\oplus) :: a \to b_1 \to b_1$ i $(\otimes) :: a \to b_2 \to b_2$ i dowolnych wartości $c_1 :: b_1$ i $c_2 :: b_2$ istnieje operacja $(\odot) :: a \to (b_1 \times b_2) \to (b_1 \times b_2)$ i wartość $c :: b_1 \times b_2$ takie, że dla dowolnej listy $x :: [a]$ jest

Problem 9 (1 p). Prove that for arbitrary operations $(\oplus) :: a \to b_1 \to b_1$ and $(\otimes) :: a \to b_2 \to b_2$, and arbitrary values $c_1 :: b_1$ and $c_2 :: b_2$ there exist such operation $(\odot) :: a \to (b_1 \times b_2) \to (b_1 \times b_2)$ and a value $c :: b_1 \times b_2$, that for every list $x :: [a]$ one has

$$\mathit{foldr}\,(\odot)\; c\; x \quad = \quad (\mathit{foldr}\,(\oplus)\; c_1\; x, \mathit{foldr}\,(\otimes)\; c_2\; x).$$

Prawo powyższe nazywa się *prawem połowienia banana*.

It is called *the banana split law*.

Grupa zaawansowana

Advanced group

Zadanie 1 (1 pkt). Udowodnij fakt dualny do faktu z zadania 7: jeśli funkcja jest jednocześnie sprawna i zlewna, to jest homomorfizmem.

Problem 1 (1 p). Prove the dual fact to the fact from Problem 7: if a function is both leftward and rightward, then it is a homomorphism.

Zadanie 2 (1 pkt). Z poprzednich zadań wynika, że sortowanie (wyspecyfikowane w postaci algorytmu *insertion sort*) jest homomorfizmem, tj. istnieje operator $\odot$ taki, że $\mathit{sort} = \mathit{hom}\,(\odot)\; (:\,[])\; []$. Pokaż, że możemy przyjąć

Problem 2 (1 p). Previous problems imply that sorting (specified in the form of the insertion sort algorithm) is a homomorphism, *i.e.*, there exitst such an operator $\odot$, than $\mathit{sort} = \mathit{hom}\,(\odot)\; (:\,[])\; []$. Show that we can take

$$u \odot v \quad = \quad \mathit{sort}\,(u +\!\!+\, v).$$

Pokaż, że

$$u \odot v \quad = \quad \mathit{foldl}\,(\mathit{flip}\ \mathit{ins})\ u\ [].$$

Pokaż następnie, że jeśli a jest nie większe od każdego elementu list x i y

$$\mathit{foldl}\,(\mathit{flip}\ \mathit{ins})\ (a : x)\ y \quad = \quad a : \mathit{foldl}\,(\mathit{flip}\ \mathit{ins})\ x\ y.$$

Oznaczmy $\mathit{foldl}\,(\mathit{flip}\ \mathit{ins})$ przez merge. Pokaż, że

$$\begin{aligned}
\mathit{merge}\ []\ y &= y \\
\mathit{merge}\ x\ [] &= x \\
\mathit{merge}\ (a : u)\ (b : v) &= \begin{cases} a : \mathit{merge}\ u\ (b : v), & a \le b, \\ b : \mathit{merge}\ (a : u)\ v, & \text{otherwise.} \end{cases}
\end{aligned}$$

Wywnioskuj stąd, że jeśli

$$\begin{aligned}
\mathit{isort} &:: \mathit{Ord}\ a \Rightarrow [a] \to [a] \\
\mathit{isort} &= \mathit{foldr}\ \mathit{insert}\ [] && (24)
\end{aligned}$$

$$\begin{aligned}
\mathit{insert} &:: \mathit{Ord}\ a \Rightarrow a \to [a] \to [a] \\
\mathit{insert}\ x\ [] &= [x] && (25) \\
\mathit{insert}\ x\ \mathit{ys}@(y : \mathit{ys}') & \\
\quad x \le y &= x : \mathit{ys} && (26) \\
\quad \mathit{otherwise} &= y : \mathit{insert}\ x\ \mathit{ys}' && (27)
\end{aligned}$$

$$\begin{aligned}
\mathit{msort} &:: \mathit{Ord}\ a \Rightarrow [a] \to [a] \\
\mathit{msort}\ [] &= [] && (28) \\
\mathit{msort}\ [x] &= [x] && (29) \\
\mathit{msort}\ \mathit{xs} &= \mathit{merge}\ (\mathit{msort}\ (\mathit{take}\ n\ \mathit{xs}))\ (\mathit{msort}\ (\mathit{drop}\ n\ \mathit{xs})) && (30) \\
&\quad \mathit{where}\ n = \mathit{length}\ \mathit{xs}\ \text{`}\mathit{div}\text{`}\ 2 && (31)
\end{aligned}$$

$$\begin{aligned}
\mathit{merge} &:: \mathit{Ord}\ a \Rightarrow [a] \to [a] \to [a] \\
\mathit{merge}\ \mathit{xs}@(x : \mathit{xs}')\ \mathit{ys}@(y : \mathit{ys}') & \\
\quad x \le y &= x : \mathit{merge}\ \mathit{xs}'\ \mathit{ys} && (32) \\
\quad \mathit{otherwise} &= y : \mathit{merge}\ \mathit{xs}\ \mathit{ys}' && (33) \\
\mathit{merge}\ \mathit{xs}\ [] &= \mathit{xs} && (34) \\
\mathit{merge}\ []\ \mathit{ys} &= \mathit{ys} && (35)
\end{aligned}$$

$$\begin{aligned}
\mathit{take} &:: \mathit{Int} \to [a] \to [a] \\
\mathit{take}\ _\ [] &= [] && (36) \\
\mathit{take}\ n\ (x : \mathit{xs}) & \\
\quad n > 0 &= x : \mathit{take}\ (n - 1)\ \mathit{xs} && (37) \\
\quad \mathit{otherwise} &= [] && (38)
\end{aligned}$$

$$\begin{aligned}
\mathit{drop} &:: \mathit{Int} \to [a] \to [a] \\
\mathit{drop}\ _\ [] &= [] && (39) \\
\mathit{drop}\ n\ \mathit{xs}@(_ : \mathit{xs}') & \\
\quad n > 0 &= \mathit{drop}\ (n - 1)\ \mathit{xs}' && (40) \\
\quad \mathit{otherwise} &= \mathit{xs} && (41)
\end{aligned}$$

to $\mathit{isort} = \mathit{msort}$.

Next show that if a is not greater than any element of the lists x and y, then

Let merge stands for $\mathit{foldl}\,(\mathit{flip}\ \mathit{ins})$. Show that:

Conclude that if

then $\mathit{isort} = \mathit{msort}$.