

TÖL304G

Forritunarmál

Verkefnablað 10

Snorri Agnarsson

23. október 2024

Verkefni — Exercises

Munið að föll sem skilað er, þar með talið hjálparföll, hafa skýra og rétta lýsingu með „Notkun: ...“, „Fyrir: ...“ og „Gildi: ...“. Takið eftir að í sumum tilfellum þurfa forskilyrði að innihalda lýsingar á sama sniði fyrir viðföng sem eru föll, og svipað gildir í eftirskilyrði (þ.e. „Gildi: ...“) fyrir gildi sem eru föll.

Remember that all functions turned in, including helper functions, should have a clear and correct description with "Use: ...", "Pre: ...", and "Value: ...". Note that in some instances preconditions need to contain similar descriptions for arguments, and similarly for the postcondition (i.e. "Value: ...") for values that are functions.

Mikilvægt er að – It is important that:

- Allar lausnir séu vel sniðsettar og *með réttri innfellingu*.

All solutions are well formatted and *correctly indented*.

- Öll **föll** hafi skýra lýsingu með „Notkun:“, „Fyrir:“ og annaðhvort „Eftir:“ eða „Gildi:“, þ.e. forskilyrði og eftirskilyrði. Sé lýsingin gefin fyrirfram skal nota hana, en athugið að hjálparföll sem þið finnið sjálf upp þurfa sína lýsingu. Athugið samt að þótt gefin sé lýsing falls má skrifa aðra lýsingu ef forskilyrði er víkkað og eftirskilyrði er þrengt. Það kemur fyrir sð slíkt sé gagnlegt, til dæmis ef endurkvæm notkun fallsins krefst betra falls en upphaflega lýsingin krefst.

All **functions** have a clear description with "Use:", "Pre:" and either "Post:" or "Value:", i.e. preconditions and postconditions. If the description is given in the exercise definition, that should be used, but note that helper functions that you invent yourself need their descriptions. However, it should be noted that

even though a description is given, you can supply another description as long as the new description has a more relaxed precondition and a more restrictive postcondition. Sometimes this is useful, for example if recursive use requires a function with an improved description.

- Sérhver nemandi skal vinna sín einstaklingsverkefni einn og óháður öðrum nemendum. Samræður milli nemenda um verkefnin eru af hinu góða, en afritun einstaklingsverkefna er að sjálfsgöðu óheimil.

Students should solve their individual exercises by themselves, independently of other students. Conversations between students are good, but copying of individual solutions is strictly forbidden.

Verkefni — Assignments

Haskell verkefni — Haskell assignments:

Hópverkefni — Group Assignments

1. Notið `foldl` eða `foldr` til að búa til Haskell fall sem tekur sem viðfang lista gilda $[x_1, x_2, \dots, x_N]$ og skilar listanum $[[x_N], \dots, [x_2], [x_1]]$. Stofninn á fallinu skal vera eitt kall á `foldl` eða `foldr` með viðeigandi viðföngum.

Use `foldl` or `foldr` to write a Haskell function that takes as argument a list $[x_1, x_2, \dots, x_N]$ of values and returns the list $[[x_N], \dots, [x_2], [x_1]]$. The body of the function should be a single call to `foldl` or `foldr` with the appropriate arguments.

2. Skriðið fall í Haskell sem tekur eitt viðfang sem er listi lista af fleytitölum og skilar margfeldinu af summunum af gildum innri listanna. Athugið að summa núll talna er 0.0 og margfeldi núll talna er 1.0. Ekki má nota innbyggðu föllin `sum` og `product` í Haskell. Hins vegar má nota föllin `foldl`, `foldr` og `map` ef þau henta.

Write a function in Haskell that takes one argument that is a list of lists of floating point numbers and returns the product of the sums of the inner list. Note that the sum of zero numbers is 0.0 and the product of zero numbers is 1.0. Using the built-in functions `sum` and `product` is not allowed. On the other hand you may use the functions `foldl`, `foldr` and `map` if that is convenient.

Einstaklingsverkefni — Individual Assignments

1. Skriðið fall `listAll` í Haskell sem tekur viðföng `i`, `n` og `f` og skilar listanum $[f(i), f(i+1), \dots, f(n)]$.

Write a function `listAll` in Haskell that takes arguments `i`, `n` and `f` and returns the list `[f(i),f(i+1),...,f(n)]`.

2. Skrifðu Haskell fall `powerList` sem tekur heiltölu `n` sem viðfang og skilar lista af heiltölulistum sem eru allir mögulegir undirlistar listans `[1, 2, ..., n]`. Þetta ætti að vera veldislistinn af `[1, 2, ..., n]` eins og rætt hefur verið í öðrum dæmum.

Write a Haskell function `powerList` that takes an integer `n` as argument and returns a list of integer lists that are all the possible sublists of the list `[1, 2, ..., n]`. This should be the powerlist of `[1, 2, ..., n]` as discussed for other assignments.