

TÖL304G

Forritunarmál

Verkefnablað 6

Snorri Agnarsson

7. október 2019

Verkefni

Lýsingar falla

Í öllum verkefnunum skulu öll föll, 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.

Prófanir falla

Munið að alltaf skal sýna einhverjar ófáfengilegar prófanir á öllum föllum sem þið skilið. Ekki þarf að sýna skjáskot af prófunum, en sýna skal hvernig prófanirnar eru forritaðar og hverjar útkomurnar eru.

Gott getur verið að ræða um prófanir ykkar á milli og á Piazza, einnig fyrir einstaklingsverkefni.

Hópverkefni

Fyrir þessi verkefni skuluð þið sækja skrána `permutations.ml-beinagrind` úr Uglunni og vista hana hjá ykkur undir nafninu `permutations.ml`. Gerið síðan viðeigandi viðbætur og sýnið forritstextannn ásamt lýsingum fyrir föllin sem þið klárið.

1. Skrifðu fall `mapreduce` í CAML, þar sem kallið `mapreduce f op u x` tekur fall `f`, tvíundaraðgerð `op`, gildi `u` og lista `x = [x1; ...; xn]` og skilar gildinu $u \oplus f(x_1) \oplus \dots \oplus f(x_n)$, þar sem $x \oplus y = (op\ x\ y)$. Fallið skal vera halaendurkvæmt og skal reikna frá vinstri til hægri.

Sýnið eftirfarandi prófun á fallinu:

```
let
  inc x = x+1
and
  add x y = x+y
in
  mapreduce inc add 0 [0;1;2;3;4];;
```

2. Klárið að forrita fallið `fromTo`. Munið að sýna prófanir.
3. Klárið að forrita fallið `insertAt`. Munið að sýna prófanir.
4. Klárið að forrita fallið `extendPermutation`. Munið að sýna prófanir.
5. Klárið að forrita fallið `length`. Munið að sýna prófanir.
6. Þegar þessu er öllu lokið, sýnið útkomuna úr segðinni

```
length (permutations 6);;
```

sem er aftast í skránni ykkar.

Einstaklingsverkefni

Skrifuðu fall `powerList` í CAML sem tekur heiltölu $n \geq 0$ sem viðfang og skilar lista af listum sem inniheldur alla mögulega lista sem eru undirlistar listans $[n; n-1; \dots; 2; 1]$ í þeim skilningi að undirlistarnir innihalda gildi úr listanum $[n; n-1; \dots; 2; 1]$ í sömu röð og í $[n; n-1; \dots; 2; 1]$, nema hvað fjarlægja má núll eða fleiri gildi úr listanum $[n; n-1; \dots; 2; 1]$.

Til dæmis myndi `powerList 0` skila `[[]]` og `powerList 2` gæti skilað `[[]; [2]; [1]; [2;1]]`, eða lista sömu lista í annarri röð.

Þið megið nota eins mörg hjálparföll eins og þið viljið og nota hvaða innbyggð föll í CAML sem ykkur hentar. Athugið að tvíundaraðgerðin `@` skeytir saman tveimur listum.