

TÖL304G Forritunarmál

Skyndipróf

Leysið 6 af eftirfarandi dæmum og a.m.k. eitt úr hverjum kafla. Ef skilað er fleiri en 6 dæmum gilda 6 bestu til einkunnar innan þessara takmarkana. Öll dæmi gilda jafnt.

Engin hjálpargögn eru leyfileg. Prófið gildir sem ein dæmaskil. Skrifðið nafn ykkar og upphafsstafi dæmakennara ykkar (GPK¹ eða HHG²) á öll úrlausnarblöð.

Kafla I - Almennt efni

1. Sýnið BNF, EBNF eða málrit fyrir mál reiknisegða með svigum, heiltölunni 1, tvíundaraðgerðunum + og – og einundaraðgerðinni –. Dæmi um löglegar formúlur eru þá t.d. 1, (1) + –1 og 1 – 1, en ekki t.d. 1 – +1.
2. Lýsið í einföldum og skiljanlegum texta máli því sem eftirfarandi BNF mállýsing skilgreinir. Sýnið einnig þrjú dæmi um strengi í málinu og þrjú dæmi um strengi (með sama stafrófi) sem ekki eru í málinu.

```
<E> ::= <E> <E>
<E> ::= ( <E> )
<E> ::= ε
```

Kafla II - Bálmótun

3. Gerið ráð fyrir földun í bálmótuðu forritunarmáli eins og myndin að neðan sýnir. Forritstextinn sýnir dæmi um slíka földun í Scheme.

Gerið grein fyrir hvar í földuninni er hægt að kalla á hvert um sig af P, Q, R og S. Gerið einnig grein fyrir hvar í földuninni er unnt að nota innri breytur í P, Q, R og S.

Földunin er þannig að P inniheldur Q og R, og R inniheldur S, þ.e.:

```

P
  Q
    R
      S
```

Forritstexti í Scheme með slíka földun:

```
(define (P ...)
  (define (Q ...)
    ...
  )
  (define (R ...)
    (define (S ...)
      ...
    )
    ...
  )
  ...
)
```

Forritstexti í Morpho með slíka földun:

```
rec fun P(...)
{
  rec
    fun Q(...)
    {
      ...
    },
    fun R(...)
    {
      rec fun S(...)
      {
        ...
      };
      ...
    };
  ...
};
```

¹Guðmundur Páll Kjartansson kennir fyrir hádegi á þriðjudögum

²Hallgrímur Heiðar Gunnarsson kennir eftir hádegi á þriðjudögum

4. Hvað er geymt í vakningarfærslum? Hver er munurinn á innihaldi vakningarfærslna í bálkmótuðum forritunarmálum, annars vegar, og hins vegar forritunarmálum sem ekki eru bálkmótuð?

Kaffi III - Fallsforritun og listavinnsla

5. Skriðið fall í Scheme eða Morpho sem tekur tvö viðföng, annars vegar einundarumsögn (unary predicate)³, og hins vegar lista gilda sem eru gild viðföng í umsögnina. Fallið skal skila nýjum lista sem er listi þeirra gilda úr upphaflega listanum sem uppfylla umsögnina, þ.e. þeirra gilda sem umsögnin er sönn fyrir. Þið ráðið röð gildanna í útkomulistanum.
6. Forritið straum í Scheme eða Morpho sem inniheldur óendanlegu rununa

[1 3 7 15 31 ...]

7. Skriðið fall `mapreduce` í Scheme, sem tekur þrjú viðföng, einundarfall f , tvíundarfall $+$ og lista $x = (x_1 x_2 \dots x_n)$, og skilar gildinu $f(x_1) + f(x_2) + \dots f(x_n)$.

Kaffi IV - Einingaforritun

8. Skriðið

- hönnunarskjal,
- fastayrðingu gagna
- og smíð (forritun) á aðgerð til að bæta við gildi

fyrir forgangsbiðraðaeiningu (priority queue) í Morpho. Athugið að biðröð og forgangsbiðröð er ekki það sama.

Athugið að aðeins þarf að forrita eina aðgerð. Athugið einnig að ekki er beðið um hraðvirka útfærslu.

Munið upplýsingahuld. Þið ráðið hvort þið notið hlutbundna forritun.

9. Skriðið eftirfarandi atriði fyrir hlaðaeiningu í Morpho:

- hönnunarskjal
- fastayrðingu gagna
- stef eða boð sem fjarlægir gildi

Athugið að aðeins þarf að forrita eina aðgerð. Þið hafið frjálssar hendur um fastayrðingu gagna.

Munið upplýsingahuld. Þið ráðið hvort þið notið hlutbundna forritun.

³Einundarumsögn er fall sem tekur eitt viðfang og skilar Boolsku gildi, þ.e. satt eða ósatt.

TÖL304G Forritunarmál

Skyndipróf

Svör

Leysið 6 af eftirfarandi dæmum og a.m.k. eitt úr hverjum kafla. Ef skilað er fleiri en 6 dæmum gilda 6 bestu til einkunnar innan þessara takmarkana. Öll dæmi gilda jafnt. Engin hjálpargögn eru leyfileg. Prófið gildir sem ein dæmaskil. Skrifðið nafn ykkar og upphafsstafi dæmakennara ykkar (GPK¹ eða HHG²) á öll úrlausnarblöð.

Kaffi I - Almennt efni

1. Sýnið BNF, EBNF eða málrit fyrir mál reiknisegða með svigum, heiltölunni 1, tvíundaraðgerðunum + og – og einundaraðgerðinni –. Dæmi um löglegar formúlur eru þá t.d. 1, (1) + –1 og 1 – 1, en ekki t.d. 1 – +1.

Svar (BNF):

$\langle \text{expr} \rangle ::= 1 \mid (\langle \text{expr} \rangle) \mid - \langle \text{expr} \rangle \mid \langle \text{expr} \rangle + \langle \text{expr} \rangle \mid \langle \text{expr} \rangle - \langle \text{expr} \rangle$

2. Lýsið í einföldum og skiljanlegum texta máli því sem eftirfarandi BNF mállýsing skilgreinir. Sýnið einnig þrjú dæmi um strengi í málinu og þrjú dæmi um strengi (með sama stafrófi) sem ekki eru í málinu.

$\langle E \rangle ::= \langle E \rangle \langle E \rangle$

$\langle E \rangle ::= (\langle E \rangle)$

$\langle E \rangle ::= \epsilon$

Svar: Þetta er mál allra strengja yfir stafrófið ‘(, ‘)’ þar sem svigar eru í jafnvægi, þ.e. allir svigar sem opnast, lokast aftur í strengnum, og ekki má loka sviga sem ekki hefur opnast frammar.

Dæmi um strengi í málinu: (), (()), (()), ϵ .

Dæmi um strengi sem ekki eru í málinu:)(, ()), (().

Kaffi II - Bálmótun

3. Gerið ráð fyrir földun í bálmótuðu forritunarmáli eins og myndin að neðan sýnir. Forritstextinn sýnir dæmi um slíka földun í Scheme.

Gerið grein fyrir hvar í földuninni er hægt að kalla á hvert um sig af P, Q, R og S. Gerið einnig grein fyrir hvar í földuninni er unnt að nota innri breytur í P, Q, R og S.

Földunin er þannig að P inniheldur Q og R, og R inniheldur S, þ.e.:

P
Q
R
S

Forritstexti í Scheme með slíka földun:

```
(define (P ...)  
  (define (Q ...)  
    ...  
  )  
)
```

```
(define (R ...)  
  (define (S ...)  
    ...  
  )  
  ...  
)
```

¹Guðmundur Páll Kjartansson kennir eftir hádegi á þriðjudögum

²Hallgrímur Heiðar Gunnarsson kennir fyrir hádegi á þriðjudögum

```
)
...
)
```

Forritstexti í Morpho með slíka földun:

```
rec fun P(...)
{
  rec
    fun Q(...)
    {
      ...
    },
    fun R(...)
    {
      rec fun S(...)
      {
        ...
      };
      ...
    };
  ...
};
```

Svar:

- Kalla má á P úr P, Q, R og S (og fyrir utan P).
 - Kalla má á Q úr P, Q, R og S.
 - Kalla má á R úr P, Q, R og S.
 - Kalla má á S úr R og S.
 - Innri breytur P má nota í P, Q, R og S.
 - Innri breytur Q má aðeins nota í Q.
 - Innri breytur R má nota í R og S.
 - Innri breytur S má aðeins nota í S.
4. Hvað er geymt í vakningarfærslum? Hver er munurinn á innihaldi vakningarfærslna í bálkmótuðum forrit-unarmálum, annars vegar, og hins vegar forritunarmálum sem ekki eru bálkmótuð?

Svar: Vakningarfærslur innihalda viðföng, vendivistfang, tengihlekk (aðgangshlekk), stýrihlekk og stað-værar breytur. Aðeins bálkmótuð forritunarmál innihalda tengihlekki, að öðru leyti er innihaldið eins fyrir bálkmótuð forritunarmál og önnur forritunarmál.

Kafi III - Fallsforritun og listavinnsla

5. Skrifðu fall í Scheme eða Morpho sem tekur tvö viðföng, annars vegar einundarumsögn (unary predicate)³, og hins vegar lista gilda sem eru gild viðföng í umsögnina. Fallið skal skila nýjum lista sem er listi þeirra gilda úr upphaflega listanum sem uppfylla umsögnina, þ.e. þeirra gilda sem umsögnin er sönn fyrir. Þið ráðið röð gildanna í útkomulistanum.

Svar:

```
;; Notkun: (filter p x)
;; Fyrir: p er einundarumsögn,
;;        x = (x1 ... xN) er listi gilda
;;        sem umsögnin p getur tekið sem
;;        viðfang.
;; Gildi: Listi þeirra gilda úr x sem
;;        uppfylla umsögnina.
```

³Einundarumsögn er fall sem tekur eitt viðfang og skilar Boolsku gildi, þ.e. satt eða ósatt.

```

(define (filter p x)
  (if (null? x)
      '()
      (if (p (car x))
          (cons (car x) (filter p (cdr x)))
          (filter p (cdr x)))
      )
  )
)

```

6. Forritið straum í Scheme eða Morpho sem inniheldur óendanlegu rununa

[1 3 7 15 31 ...]

Svar:

```

;; Notkun: (transform s)
;; Fyrir: s er óendanlegur straumur [x1 x2 ...]
;; Gildi: Straumurinn [2*x1+1 2*x2+1 ...]
(define (transform s)
  (cons-stream (+ 1 (* 2 (stream-car s))) (transform (stream-cdr s)))
)

;; Notkun: svar
;; Gildi: Straumurinn [1 3 7 15 31 ...]
(define svar (cons-stream 1 (transform svar)))

```

7. Skriðið fall `mapreduce` í Scheme, sem tekur þrjú viðföng, einundarfall f , tvíundarfall $+$ og lista $x = (x_1 x_2 \dots x_n)$, og skilar gildinu $f(x_1) + f(x_2) + \dots + f(x_n)$.

Svar:

```

;; Notkun: (mapreduce f + x)
;; Fyrir: x er listi (x1 ... xN), ekki tómur,
;;        + er tvíundarfall, f er einundarfall.
;; Gildi: f(x1)+f(x2)+...+f(xN)
(define (mapreduce f + x)
  (if (null? (cdr x))
      (f (car x))
      (+ (f (car x)) (mapreduce f + (cdr x)))
  )
)

```

Kafli IV - Einingaforritun

8. Skriðið

- hönnunarskjal,
- fastayrðingu gagna
- og smíð (forritun) á aðgerð til að bæta við gildi

fyrir forgangsbiðraðaeiningu (priority queue) í Morpho. Athugið að biðröð og forgangsbiðröð er ekki það sama.

Athugið að aðeins þarf að forrita eina aðgerð. Athugið einnig að ekki er beðið um hraðvirka útfærslu.

Munið upplýsingahuld. Þið ráðið hvort þið notið hlutbundna forritun.

Svar:

```

;;; Útflutt
;;; =====
;;; Notkun: p = makePQ();
;;; Eftir: p er ný tóm forgangsbiðröð
;;;
;;; Notkun: add(p,x);
;;; Fyrir: p er forgangsbiðröð,
;;;        x er gildi sem er löglegt viðfang í innfluttu
;;;        samanburðaraðgerðina <==.
;;; Eftir: Búið er að bæta x í p.
;;;
;;; Notkun: x = remove(p);
;;; Fyrir: p er forgangsbiðröð, ekki tóm.
;;; Eftir: x er það gildi sem var fremst í p, m.v. innfluttu
;;;        samanburðaraðgerðina <==.
;;;
;;; Notkun: b = isEmpty(p);
;;; Fyrir: p er forgangsbiðröð.
;;; Eftir: b er satt ef p er tóm, ósatt annars.
;;;
;;; Innflutt
;;; =====
;;; Notkun: b = x <== y;
;;; Fyrir: x og y eru gildi af þeirri gerð sem við viljum
;;;        geta geymt í forgangsbiðröð.
;;; Eftir: b er satt ef x verður að vera á undan y, ósatt
;;;        ef x má ekki vera á undan y.

"pq.mmod" =
{{
;;; Fastayrðing gagna:
;;;   Forgangsbiðröð sem inniheldur gildi x1, x2, ..., xN
;;;   þar sem x1 má vera á undan x2, o.s.frv., er geymd
;;;   sem listi [[x1,x2,...,xN]].

makePQ =
  fun()
  {
    [[]];
  };

add =
  fun(p,x)
  {
    ;;; Notkun: g = insert(h,x);
    ;;; Fyrir: h er listi [x1,...,xN] af gildum sem
    ;;;        <== ræður við, þ.a. x1 má vera á undan
    ;;;        x1 o.s.frv., x er einnig gildi sem <==
    ;;;        ræður við.
    ;;; Eftir: g er listi [y1,...,yM] sem inniheldur
    ;;;        x auk x1,...,xN, þ.a. y1 má vera á undan
    ;;;        y2 o.s.frv.
    rec fun insert(h,x)
    {
      h==[] && (return [x]);
      (x <== head(h)) && (return x:h);
    }
  }

```

```

        head(h) : insert (tail (h) , x) ;
    };
    setHead (p, insert (head (p) , x) ) ;
};

remove =
    fun (p)
    {
        val res = head (head (p) ) ;
        setHead (p, tail (head (p) ) ) ;
        res ;
    };

isEmpty =
    fun (p)
    {
        head (p) == [] ;
    };
}}
;

```

9. Skrifðu eftirfarandi atriði fyrir hlaðæiningu í Morpho:

- hönnunarskjal
- fastayrðingu gagna
- stef eða boð sem fjarlægir gildi

Athugið að aðeins þarf að forrita eina aðgerð. Þið hafið frjálssar hendur um fastayrðingu gagna.

Munið upplýsingahuld. Þið ráðið hvort þið notið hlutbundna forritun.

Svar: Sjá vikublöð.