

TÖL304G Miðmisserispróf – TÖL304G Midterm Exam

1. Öll hjálpargögn eru leyfileg. All help materials are allowed.
2. Þið megið ekki tala við aðra um prófið eða skyld efni fyrr en próftíma lýkur. You must not communicate with others about the exam or related issues until the exam ends.
3. Þið megið senda private spurningar til kennara í Piazza meðan á próftíma stendur. You may ask private questions to teachers on Piazza while the exam lasts.
4. Hægt verður að ræða við mig einslega á Zoom u.þ.b. kl. 10:30 og kl. 11:15. I will be available for private consultation on Zoom at approximately 10:30 and 11:15.
5. Skilið lausn prófsins í Gradescope á sama hátt og heimaverkefnum er skilað. Turn in the solution to the exam into Gradescope in the same manner you turn in home assignments.
6. Próftíminn er frá 10:00 til 12:30. The exam lasts from 10:00 till 12:30.
7. Ef lausninni er skilað í Gradescope eftir kl. 12:30 er einn heill dreginn frá einkunn fyrir hverjar byrjaðar 10 mínútur sem farið er yfir skilafrest. If the solution is turned in to Gradescope after 12:30 there will be a grade reduction of one (out of 10) for each started 10 minutes over the deadline

Með því að skila inn þessu prófi staðfesti ég að þetta próf er unnið af mér og að ég hef hvorki veitt né fengið aðstoð sem er óleyfileg í vinnu við þetta próf. Ég skil að brot á reglum Háskóla Íslands geta haft í för með sér þung viðurlög, allt frá því að hafa áhrif á einkunn í viðkomandi námskeiði upp í brottrekstur úr skóla, tímabundið eða fyrir fullt og allt.

By turning in this exam I confirm that this exam was solved by me and that I have neither given nor accepted any help that is not allowed in solving this exam. I understand that breaking the rules of the University of Iceland can have serious consequences including grade reduction and expulsion from the school, temporarily or permanently.

1. (5%) Sýnið BNF, EBNF eða málrít fyrir mál segða yfir stafrófið $\{x, +, (,)\}$. Svigar verða að vera í jafnvægi, $+$ er tvíundaraðgerð og x er breytunafn, sem er eina leyfða frumstæða segðin.

Show BNF, EBNF or syntax diagrams for the language of expressions over the alphabet $\{x, +, (,)\}$. Parentheses must be balanced, $+$ is a binary operation, and x is a variable name, which is the only allowed primitive expression.

Dæmi um strengi í málinu. Examples of strings in the language.

x
 $x + x + x$
 (x)
 $((x)))$
 $(x) + (x)$
 $x + (x + x + (x + x) + x) + x + x$

Dæmi um strengi ekki í málinu. Examples of strings not in the language.

ϵ (tómi strengurinn, the empty string)
(
)
 $+x$
 $((x)$
 xx
 y

Svar/Answer:

EBNF: $e = 'x' \mid e, '+' , e \mid '(', e, ')'$

2. (10%)

- a. Íhugið eftirfarandi EBNF skilgreiningu fyrir mál yfir stafrófið $\{ (,), a \}$.

Consider the following EBNF definition for a language over the alphabet $\{ (,), a \}$.

$$e = 'a' \mid e, '(', e, ')';$$

Hverjir eftirfarandi strengja eru í málinu sem skilgreint er? Tvö röng svör gefa núll stig. Í svarinu skuluð þið telja upp þá liði sem eru sannir í vaxandi röð.

Which of the following strings are in the language defined? Two wrong answers give zero points. In the answer you should enumerate the correct items in ascending order.

- | | |
|--------------|--------------|
| 1. $a(a(a))$ | 6. (a) |
| 2. $a(a)(a)$ | 7. $a(a)$ |
| 3. a | 8. $a)a($ |
| 4. $($ | 9. $a()$ |
| 5. $)$ | 10. $a(a)()$ |

- b. Hverjar af eftirfarandi reglulegu segðum skilgreina eitthvert mál yfir stafrófið $\{a, b\}$ sem er undirmengi málsins sem EBNF skilgreiningin lýsir? Tvö röng svör gefa núll stig. Í svarinu skuluð þið telja upp þá liði sem eru sannir í vaxandi röð.

Which if the following regular expressions define some language over the alphabet $\{a, b\}$ which is a subset of the language defined by the given EBNF? Two wrong answers give zero points. In the answer you should enumerate the correct items in ascending order.

$$\text{EBNF: } e = 'a', e, 'b', e \mid ;$$

1. a^*b^*
2. $(a(a(ab)^*b)^*b)^*$
3. $aa(aabb)^*bbab$
4. $(ab|a|b)^*$
5. $(ab)^*$
6. $a(bbaa)^*b$
7. $a(ba)^*b$
8. $(ab|a(ab)^*b)^*$

Svar/Answer

a) 1,2,3,7

b) 2,3,5,7,8

3. (15%)

Hverjar eftirfarandi fullyrðinga um lokanir eru sannar? Tvö röng svör gefa núll punkta. Teldu upp þá punkta sem eru sannir í stafrófsröð.

Which of the following statements about closures are true? Two wrong answers give zero points. Enumerate the items that are correct in alphabetic order.

- a) Lokanir eru til í C. Closures exist in C.
- b) Lokanir eru til í Scheme. Closures exist in Scheme.
- c) Lokanir eru til í CAML. Closures exist in CAML.
- d) Lokanir eru til í Morpho. Closures exist in Morpho.
- e) Lokanir eru nauðsynlegar til að skila staðværu falli sem gildi í bálmótuðum forritunarmálum. Closures are necessary in order to return a local function as a value in block structured programming languages.
- f) Lokanir innihalda straum. Closures contain a stream.
- g) Lokanir innihalda fallsbendi. Closures contain a function pointer.
- h) Lokanir má nota til að útfæra strauma í Scheme. Closures can be used to implement streams in Scheme.
- i) Lokanir innihalda tengihlekk (aðgangshlekk). Closures contain an access link (static link).
- j) Lokanir innihalda stýrihlekk. Closures contain a control link (dynamic link).
- k) Lokanir eru aðeins mögulegar ef vakningarfærslur eru í kös. Closures are only possible if activation records are in the heap.

Svar/Answer: b,c,d,e,g,h,i

- 4 . (15%)** Skrifðið **halaendurkvæmt** fall í Scheme, CAML eða Morpho sem tekur lista heiltalna x_1, x_2, \dots, x_n sem sitt eina viðfang og skilar $\sum_{i=1}^n x_i^3$. Ekki má nota lykkju og ekki má gefa breytu nýtt gildi. Þið þurfið hjálparfall sem þið þurfið að skrifa **ásamt lýsingu þess og lýsingu þessa falls**. Aðeins má nota einfaldar innbyggðar aðgerðir í Scheme, CAML og Morpho, þ.e. aðgerðir sem hafa tímaflækju $O(1)$ svo sem cons, car, cdr, null?, + og *. Munið að meðhöndla tilvikið þegar $n = 0$. Write a **tail recursive** function in Scheme, CAML or Morpho, which takes a list of integers x_1, x_2, \dots, x_n as its only argument and returns $\sum_{i=1}^n x_i^3$. No loops are allowed and it is not allowed to give a variable a new value. You will need a helper function, which you must write yourself, **along with its descriptions as well as the description of this function**. Only simple built-in operations in Scheme, CAML and Morpho may be used, i.e. operations that have time complexity $O(1)$ such as cons, car, cdr, null?, +, and *. Remember to handle the case when $n = 0$.

Svar/Answer:

```
;; Use:      (sumcubes x)
;; Pre:      x=(x1 x2 ... xN) is a list of integers
;; Value:    x1^3+x2^3+...+xN^3
(define (sumcubes x)
  ;; Use:      (help x s)
  ;; Pre:      x=(x1 x2 ... xN) is a list of
  ;;           numbers, s is a number
  ;; Value:    s+x1^3+x2^3+...+xN^3
  (define (help x s)
    (if (null? x)
        s
        (help (cdr x)
                (+ s
                   (* (car x) (car x) (car x))
                )
        )
    )
  (help x 0)
)
```

5. (15%)

Gerið ráð fyrir eftirfarandi Scheme föllum eða samsvarandi í CAML eða Morpho.

Assume the following Scheme functions or their equivalents in CAML or Morpho.

```
;; Notkun: (split x)
;; Fyrir:  x=(x1 ... xN) er listi einhverra gilda
;; Gildi:  Listi tveggja lista (y z) þannig að y
;;         og z saman innihalda öll gildin í x
;;         eins oft og x inniheldur þau og
;;         helmingurinn er í y og hinn
;;         helmingurinn er í z (það má muna einum
;;         í fjölda).
;; Ath.:   Það er ekkert loforð um að tvö köll
;;         með sama viðfangi skili sömu
;;         niðurstöðu
```

```
;; Usage:  (split x)
;; Pre:     x=(x1 ... xN) is a list of some values
;; Value:   A list of two lists (y z) such that y
;;         and z together contain all the values
;;         in x as many times as x contains them
;;         and half are in y and half are in z
;;         (the count of values may differ by
;;         one).
;; Note:    There is no guarantee that two calls
;;         with the same parameter gives the same
;;         result
```

```
;; Notkun: (merge x y)
;; Fyrir:  x og y eru talnalistar í vaxandi röð
;; Gildi:  Listi allra talnanna í x og y í
;;         vaxandi röð
```

```
;; Usage:  (merge x y)
;; Pre:     x and y are lists of numbers in
;;         ascending order
;; Value:   A list of all the numbers in x and y
;;         in ascending order
```

Skrifið fall í Scheme, CAML eða Morpho til að raða heiltalnalistum af

hvaða lengd sem er. Notið aðeins einfaldar aðgerðir (auk split og merge) og engar lykkjur eins og í dæminu á undan. Munið lýsingar falla.

Write a function in Scheme, CAML or Morpho to sort integer lists of any length. Use only simple operations (plus split and merge) and no loops as in the previous question. Remember function descriptions.

Svar/Answer:

```
;; Use:      (sort x)
;; Pre:      x is a list of integers
;; Value:    A list of the same integers in
;;           ascending order
(define (sort x)
  (if (or (null? x) (null? (cdr x)))
      x
      ((lambda (s)
         (merge (sort (car s)) (sort (cadr s)))
        )
       (split x)
      )
  )
)
```

6. (15%)

Skrifið halaendurkvæmt fall `sumFromTo` í Scheme, CAML eða Morpho, sem tekur viðföng i , j og f , þar sem i og j eru heiltölur og f er fall sem tekur heiltölu sem viðfang og skilar fleytitölu. Kallið á `sumFromTo` skal skila $\sum_{i \leq k \leq j} f(k)$. Munið að meðhöndla tilvik þegar $i > j$. Munið að lýsa öllum föllum með Notkun/Fyrir/Gildi. Notið aðeins einfaldar aðgerðir og engar lykkjur eins og í dæminu á undan.

Write a tail recursive function `sumFromTo` in Scheme, CAML or Morpho, that takes arguments i , j and f , where i and j are integers and f is a function that takes an integer as argument and returns a floating point number. The call to `sumFromTo` should return $\sum_{i \leq k \leq j} f(k)$. Remember to handle cases where $i > j$. Remember to describe all functions with Use/Pre/Value. Use only simple operations and no loops as in the previous question.

Svar/Answer:

```
;; Use:      (sumFromTo f i j)
;; Pre:      i and j are integers, f is a function that
;;           takes an integer as argument and returns a
;;           floating point number
;; Value:    The sum of all (f k) for all k such that
;;           i <= k <= j
(define (sumFromTo f i j)
  ;; Use:      (help s f i j)
  ;; Pre:      i and j are integers, f is a function
  ;;           that takes an integer as argument and
  ;;           returns a floating point number, s is
  ;;           a floating point number
  ;; Value:    s plus the sum of all (f k) for all
  ;;           k such that i <= k <= j
  (define (help s f i j)
    (if (> i j)
        s
        (help (+ (f i) s) (+ i 1) j)
    )
  )
  (help 0.0 f i j)
)
```


7. (15%)

Skrifið fall `transpose` í Scheme eða Morpho sem hefur eftirfarandi lýsingu í Scheme.

Write a function `transpose` in Scheme or Morpho that has the following description in Scheme.

```
;; Notkun: (transpose s)
;; Fyrir: s er óendanlegur straumur óendanlegra
;;        strauma:
;;        s=[x11 x12 x13 ...]
;;           [x21 x22 x23 ...]
;;           [x31 x32 x33 ...]
;;           ...
;;        ]
;; Gildi: Óendanlegi straumurinn sem er
;;        byltingin (transpose) af s:
;;        [[x11 x21 x31 ...]
;;         [x12 x22 x32 ...]
;;         [x13 x23 x33 ...]
;;         ...
;;        ]

;; Usage: (transpose s)
;; Pre:   s is an infinite stream of infinite
;;        streams::
;;        s=[x11 x12 x13 ...]
;;           [x21 x22 x23 ...]
;;           [x31 x32 x33 ...]
;;           ...
;;        ]
;; Value: The infinite stream that is the
;;        transpose of s:
;;        [[x11 x21 x31 ...]
;;         [x12 x22 x32 ...]
;;         [x13 x23 x33 ...]
;;         ...
;;        ]
```

Vísbending: Í Scheme megið þið reikna með að allar okkar straumaaðgerðir séu til staðar og í Morpho megið þið reikna með að til sé fall `streamMap` sem virkar svipað og samsvarandi straumafall í Scheme. Athugið að til að senda fall `f` sem viðfang í Morpho þarf að senda lokun svo sem `fun(x) { f(x) }`.

Hint: In Scheme you may assume the existence of all our stream operations and in Morpho you may assume the existence of a function

streamMap that works similarly to the corresponding Scheme function.
 Note that to send a function f as an argument in Morpho you need to send a closure such as $\text{fun}(x)\{f(x)\}$.

Svar/Answer:

```
;; Usage: (transpose s)
;; Pre:   s is an infinite stream of infinite
;;        streams::
;;        s=[[x11 x12 x13 ...]
;;           [x21 x22 x23 ...]
;;           [x31 x32 x33 ...]
;;           ...
;;        ]
;; Value: The infinite stream that is the
;;        transpose of s:
;;        [[x11 x21 x31 ...]
;;         [x12 x22 x32 ...]
;;         [x13 x23 x33 ...]
;;         ...
;;        ]
(define (transpose s)
  (cons-stream
    (stream-map stream-car s)
    (transpose (stream-map stream-cdr s)))
)
```

8. (10%)

Íhugið eftirfarandi Scheme fall – Consider the following Scheme function.

```
(define (P a)
  (define b 1)
  (define (Q c)
    (define d 2)
    (define (R e)
      (define f 3)
      (+ a b c d e f))
    R)
  )
  (define (S i)
    (i 4))
  )
  (S (Q 5)))
```

Gerið ráð fyrir að kallað sé á (P 0). Inni í því kalli verður kallað á R. Lýsið innihaldi vakningarfærslunnar fyrir R (teljið upp allt sem hún inniheldur) og gerið grein fyrir eftirfarandi atriðum í vakningarfærslunni:

1. Viðföngum í vakningarfærslunni, nöfnum og innihaldi þeirra
2. Staðværum breytum í vakningarfærslunni, nöfnum og innihaldi þeirra
3. Tengihlekk (aðgangshlekk) og á hvaða vakningarfærslu hann vísar
4. Stýrihlekk og á hvaða vakningarfærslu hann vísar

Gerið einnig á sama hátt grein fyrir vakningarfærslunni sem tengihlekkur R vísar á.

Assume (P 0) is called. Inside that call R will be called. Describe the contents of the activation record for R (enumerate everything it contains) and describe the following items in the activation record:

1. Arguments in the activation record, their names and contents
2. Local variables in the activation record, their names and contents
3. The access link (static link) and to what activation record it refers
4. The control link (dynamic link) and to what activation record it refers to

In the same way describe the contents of the activation record that the access link of R refers to.

Svar/Answer:

The activation record for R will contain:

1. The argument named e with the value 4.
2. The local variable named f with the value 3.
3. An access link that refers to an activation record for Q
4. A control link that refers to the activation record for the caller that will receive the result from the call to P (absent tail recursion this will be the activation record of the caller of P) along with a return address to that same function. (If you answer that the control link refers an activation record for S, then I will not subtract any points, because if Scheme did not handle tail recursion this would be the case)

The activation record for Q that the access link from R refers to will contain:

1. The argument named c with the value 5.
2. The local variable d with the value 2. The local variable R with a value that is a closure referring to the function R. (If you do not mention the local variable R, I will not subtract any points.)
3. An access link that refers to an activation record for P
4. A control link that may refer to an activation record for P (since the call has already finished, this is not necessary), along with a return address in P.