

# TÖL304G Lokapróf

## TÖL304G Final Exam

1. Leyfileg hjálpargögn er ein A4 blaðsíða að eigin vali skrifuð báðu megin.

Allowed help materials are one A4 page of your own choice with writing on both sides.

2. Skrifið svörin á þessar síður, ekki í prófbókina.

Write your answers on these pages, not in the exam book.

3. Ef svarið kemst ekki fyrir á svæðinu sem ætlað er fyrir það má skrifa í auðu blaðsíðurnar aftast, en þá skal segja frá því í því svæði sem ætlað er fyrir svarið, til dæmis „sjá framhald á síðu 20“.

If the answer does not fit on the allotted space you may write on the empty pages at the end, but in that case you should write an indication to that effect on the space allotted for the answer, for example “continued on page 20”.

4. Forðist að krumpa eða rífa þessar blaðsíður, þær þurfa að fara gegnum skanna. Skrifið skýrt með dökku lettri og ekki skrifa í jaðrana.

Refrain from mutilating or tearing these pages, they have to go through a scanner. Write clearly with dark letters and do not write in the margins.

Nafn/Name: .....

**(Skrifið nafn ykkar bæði hér og á forsíðu prófbókar)**

**(Write your name both here and on the cover of the exam book)**

Háskólatölvupóstfang/University Email: .....

**1. (10%)**

Svarið tveimur af eftirfarandi þremur spurningum.

Answer two of the following three questions.

- a. Sýnið BNF, EBNF eða málrit fyrir mál strengja yfir stafrófið  $\{a, +, (, )\}$  sem eru löglegar segðir með breytunafninu  $a$ , svigum og tvíundaraðgerðinni  $+$ .

Show BNF, EBNF or syntax diagrams for the language of strings over the alphabet  $\{a, +, (, )\}$ , that are legal expressions with the variable name  $a$ , parentheses and the binary operation  $+$ .

Dæmi um strengi í málinu.

Examples of strings in the language.

$a$   
 $(a)$   
 $a + a + a$   
 $a + (a)$   
 $((a)))$   
 $a + (a + a + (a + a) + a) + a + a$

Dæmi um strengi ekki í málinu.

Examples of strings not in the language.

$($   
 $a)$   
 $\epsilon$  (tómi strengurinn)  
 $((a)$   
 $aa$   
 $a +$   
 $+ a$

**Svar/Answer:**

$\langle E \rangle ::= a \mid ( \langle E \rangle ) \mid \langle E \rangle + \langle E \rangle$

- b. Íhugið eftirfarandi BNF skilgreiningu. Consider the following BNF definition.

$$\langle E \rangle ::= a \langle E \rangle b \mid \langle E \rangle p \langle E \rangle \mid c$$

Hverjir eftirfarandi strengja eru í málinu sem skilgreint er? Tvö röng svör gefa núll stig. Vísbending:  $a$  og  $b$  eru í jafnvægi.

Which of the following strings are in the language defined? Two wrong answers give zero points. Hint:  $a$  and  $b$  are in balance.

- |            |               |
|------------|---------------|
| 1. $a$     | 6. $bcac$     |
| 2. $b$     | 7. $cpc$      |
| 3. $c$     | 8. $cpacb$    |
| 4. $acb$   | 9. $acbpacb$  |
| 5. $acbpc$ | 10. $aacbbpc$ |

**Svar/Answer:**

1	2	3	4	5	6	7	8	9	10
		X	X	X		X	X	X	X

- c. Hverjar af eftirfarandi reglulegu segðum skilgreina mál þar sem fjöldi  $a$  og  $b$  er sá sami. Eitt rangt svar gefur núll stig.

Which of the following regular expressions define a language where the number of  $a$ 's and  $b$ 's are equal? One wrong answer gives zero points.

- $a^*b^*$
- $(ab)^*$
- $b(ab)^*a$
- $a(bbaa)^*b$
- $a(ba)^*b$
- $(a(b(ab)^*a)^*b)^*$
- $aa(aabb)^*bbab$
- $(ab|a|b)^*$
- $(ab|a(ab)^*b)^*$

**Svar/Answer:**

1	2	3	4	5	6	7	8	9
	X	X	X	X	X	X		X

**2. (10%)**

Hverjar eftirfarandi fullyrðinga um lokanir eru sannar? Tvö röng svör gefa núll punkta.

Which of the following statements about closures are true? Two wrong answers give zero points.

- a) Lokanir innihalda fallsbendi. Closures contain a function pointer.
- b) Lokanir má nota til að útfæra strauma í Scheme. Closures can be used to implement streams in Scheme.
- c) Lokanir innihalda tengihlekk (aðgangshlekk). Closures contain an access link (static link).
- d) Lokanir innihalda stýrihlekk. Closures contain a control link (dynamic link).
- e) Lokanir innihalda straum. Closures contain a stream.
- f) 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.
- g) Lokanir eru til í C. Closures exist in C.
- h) Lokanir eru til í Scheme. Closures exist in Scheme.
- i) Lokanir eru til í CAML. Closures exist in CAML.
- j) Lokanir eru til í Morpho. Closures exist in Morpho.
- k) Lokanir eru til í Java. Closures exist in Java.
- l) Lokanir eru aðeins mögulegar ef vakningarfærslur eru í kös. Closures are only possible if activation records are in the heap.

**Svar/Answer:**

a	b	c	d	e	f	g	h	i	j	k	l
X	X	X			X		X	X	X		

**3. (10%)**

Skrifið eftirfarandi Scheme fall. Notið aðeins einfaldar Scheme aðgerðir eins og í dæminu á eftir. **Ekki er leyfilegt að nota hjálparföll.**

Write the following Scheme function. Use only simple operations as in the next question. **It is not allowed to use helper functions.**

```
;; Notkun: (foldr f u x)
;; Fyrir:  f er tviundarfall,
;;         u er eitthvert gildi,
;;         x=(x1 ... xN) er listi einhverra gilda
;; Gildi:  (f x1 (f x2 ... (f xN u)...))
;; Ath.:   (foldr f u ()) skal skila u

;; Usage:  (foldr f u x)
;; Pre:    f is a binary function,
;;         u is some value,
;;         x=(x1 ... xN) is a list of some values
;; Value:  (f x1 (f x2 ... (f xN u)...))
;; Note:   (foldr f u ()) should return u
```

**Svar/Answer:**

```
(define (foldr f u x)
  (if (null? x)
      u
      (f (car x) (foldr f u (cdr x)))
  )
)
```

**4. (10%)**

Skrifið **halaendurkvæmt** fall í Scheme, CAML eða Morpho sem (í Morpho) hefur eftirfarandi lýsingu. Ekki má nota lykkju og ekki má gefa breytu nýtt gildi. Nota má hjálparföll sem þið þurfið þá að skrifa sjálf **ásamt lýsingu þeirra**. Aðeins má nota einfaldar innbyggðar aðgerðir í Scheme, CAML og Morpho, þ.e. aðgerðir sem hafa tímaflækju  $O(1)$  svo sem cons, car, cdr og null?.

Write a **tail recursive** function in Scheme, CAML or Morpho, which (in Morpho) has the following description. No loops are allowed and it is not allowed to give a variable a new value. Helper functions can be used, which you must then write yourself, **along with their descriptions**. 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 and null?.

```
;;; Notkun: x = atoi(n);
;;; Fyrir:  n er heiltala, n>=0.
;;; Gildi:  x er listinn [n,n-1,...,2,1]

;;; Usage:  x = atoi(n);
;;; Pre:    n is an integer, n>=0.
;;; Value:  x is the list [n,n-1,...,2,1]
```

**Svar/Answer:**

```
(define (atoi n)
  ;; Notkun: (help x i)
  ;; Fyrir:  x=(i i-1 ... 2 1) er listi,
  ;;        i er heiltala, 0<=i<=n.
  ;; Gildi:  Listinn (n n-1 ... 2 1)
  (define (help x i)
    (if (= i n)
        x
        (help (cons (+ i 1) x) (+ i 1))
    )
  )
  (help '() 0)
)
```



**5. (10%)**

Skrifið eftirfarandi einundarfall, mapper, í Scheme, CAML, Haskell eða Morpho. Lýsingin í Scheme er eftirfarandi. Notið aðeins einfaldar aðgerðir eins og í fyrra dæmi. Þið þurfið ekki að skrifa lýsingu fallsins, en ef þið notið hjálparföll þurfið þið að lýsa þeim.

Write the following unary function, mapper, in Scheme, CAML, Haskell or Morpho. The description in Scheme is the following. Use only simple operations as in the previous question. You do not need to write the description of the function, but if you use helper functions you must describe them.

```
;; Notkun: (mapper f)
;; Fyrir:  f er einundarfall
;; Gildi:  Fall g með eftirfarandi lýsingu:
;;         Notkun: (g x)
;;         Fyrir:  x=(x1 ... xN) er listi.
;;         Gildi:  Listinn (y1 ... yN) þar sem
;;                 yI er útkoman úr kallinu
;;                 (f xI), fyrir I=1,...,N

;; Usage: (mapper f)
;; Pre:   f is a unary function
;; Value: A function g with the following
;;         description:
;;         Usage: (g x)
;;         Pre:   x=(x1 ... xN) er listi.
;;         Value: The list (y1 ... yN) where
;;                 yI is the result of the call
;;                 (f xI), for I=1,...,N
```

**Svar/Answer:****Í Haskell:**

```
let mapper f x = if x==[] then [] else (f (head x)) : (mapper f (tail x))
EDA
mapper f [] = []
mapper f (x:xs) = (f x):(mapper f xs)
```



**Í Scheme:**

```
(define (mapper f)
  (lambda (x) (if (null? x) () (cons (f (car x)) ((mapper f) (cdr x)))))
)
```

**6. (10%)**

Hverjar af eftirfarandi fullyrðingum eru í samræmi við meginregluna um upplýsingahuld? Það gætu verið núll, ein eða fleiri. Tvö röng svör gefa núll stig.

Which of the following statements are in accordance with the principle of information hiding? There might be zero, one or more. Two wrong answers give zero points.

- a. Gefa skal notendum einingar fullkomnar upplýsingar um smíð einingarinnar.

The users of a module should be given perfect information about the implementation of the module.

- b. Fastayrðing gagna einingar skal vera fullkomlega aðgengileg smiðum einingarinnar.

The data invariant of a module should be perfectly accessible to the implementors of the module.

- c. Fastayrðing gagna einingar skal vera fullkomlega aðgengileg notendum einingarinnar.

The data invariant of a module should be perfectly accessible to the users of a module.

- d. Fastayrðing gagna einingar skal ekki vera aðgengileg smiðum einingarinnar.

The data invariant of a module should not be accessible to the implementors of the module.

- e. Fastayrðing gagna einingar skal ekki vera aðgengileg notendum einingarinnar.

The data invariant of a module should not be accessible to the users of the module.

- f. Smiðir einingar geta breytt fastayrðingu gagna einingarinnar. The implementors of a module can change the data invariant of the module.

- g. Notendur einingar geta breytt fastayrðingu gagna einingarinnar. The users of a module can change the data invariant of the module.

**Svar/Answer:**

a	b	c	d	e	f	g
	X			X	X	

**7. (10%)**

Útfærið, að hluta, einingu fyrir fjölnota forgangsbiðröð í Morpho. Sýnið eftirfarandi.

Implement, partially, a module for a generic priority queue in Morpho. Show the following.

- a. Hönnunarskjal sem inniheldur lýsingar (notkun/fyrir/eftir) fyrir öll innflutt og útflutt atriði einingarinnar.

A design document that contains descriptions (usage/pre/post) for all items imported to or exported from the module.

- b. Smíð einingarinnar, þar sem sleppa má útfærslu allra aðgerða nema þeirri sem fjarlægir gildi úr forgangsbiðröðinni. Athugið að sýna þarf fastayrðingu gagna.

An implementation of the module, where you can skip implementing all operations except the operation to remove a value from the priority queue. Note that you must show a data invariant.

Unnt skal vera að nota einingaraðgerðir til að búa til afbrigði af einingunni sem gefa forgangsbiðraðir fyrir hvaða gildi sem er sem hafa viðeigandi samanburðarfall. Þið ráðið hvort forgangsbiðröðin er útfærð sem hlutur eða ekki.

It should be possible to use module operations to construct a variant of the module that gives a priority queue for any values that have the appropriate comparison function. You may choose whether the priority queue is implemented as an object or not.

**Svar/Answer:**

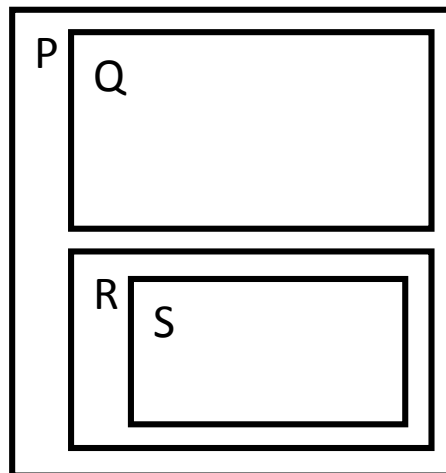
Sjá heimaverkefni.



**8. (10%)**

Íhugið myndina sem sýnir földun falla P, Q, R og S. Samsvarandi Scheme forritstexti er einnig sýndur í tveimur jafngildum útgáfum hlið við hlið.

Consider the figure which shows nesting of functions P, Q, R, and S. The corresponding Scheme code is also shown in two equivalent versions side by side.



```
(define (P ...)
  (define (Q ...)
    ...[stofn Q/body of Q]
  )
  (define (R ...)
    (define (S ...)
      ...[stofn S/body of S]
    )
    ...[stofn R/body of R]
  )
  ...[stofn P/body of P]
)
```

```
(define (P ...)
  (define (R ...)
    (define (S ...)
      ...[stofn S/body of S]
    )
    ...[stofn R/body of R]
  )
  (define (Q ...)
    ...[stofn Q/body of Q]
  )
  ...[stofn P/body of P]
)
```

Krossið í töflunni að neðan við þær af eftirfarandi fullyrðingum sem eru sannar. Eitt rangt svar gefur núll fyrir dæmið.

Make a tick mark in the table below for each of the following statements that are true. One wrong answer gives a zero grade for the problem.

- a. Kalla má á P úr Q.  
P may be called from Q.
- b. Kalla má á P úr R.  
P may be called from R.
- c. Kalla má á P úr S.  
P may be called from S.
- d. Kalla má á Q úr P.  
Q may be called from P.
- e. Kalla má á Q úr R.  
Q may be called from R.
- f. Kalla má á Q úr S.  
Q may be called from S.
- g. Kalla má á R úr P.  
R may be called from P.
- h. Kalla má á R úr Q.  
R may be called from Q.
- i. Kalla má á R úr S.  
R may be called from S.
- j. Kalla má á S úr P.  
S may be called from P.
- k. Kalla má á S úr Q.  
S may be called from Q.
- l. Kalla má á S úr R.  
S may be called from R.
- m. Staðværar breytur í P eru nothæfar í stofni Q.  
Local variables of P are usable in the body of Q.
- n. Staðværar breytur í P eru nothæfar í stofni R.  
Local variables of P are usable in the body of R.
- o. Staðværar breytur í P eru nothæfar í stofni S.  
Local variables of P are usable in the body of S.
- p. Staðværar breytur í Q eru nothæfar í stofni P.  
Local variables of Q are usable in the body of P.
- q. Staðværar breytur í Q eru nothæfar í stofni R.  
Local variables of Q are usable in the body of R.
- r. Staðværar breytur í Q eru nothæfar í stofni S.  
Local variables of Q are usable in the body of S.
- s. Staðværar breytur í R eru nothæfar í stofni P.  
Local variables of R are usable in the body of P.

- t. Staðværar breytur í R eru nothæfar í stofni Q.  
Local variables of R are usable in the body of Q.
- u. Staðværar breytur í R eru nothæfar í stofni S.  
Local variables of R are usable in the body of S.
- v. Staðværar breytur í S eru nothæfar í stofni P.  
Local variables of S are usable in the body of P.
- w. Staðværar breytur í S eru nothæfar í stofni Q.  
Local variables of S are usable in the body of Q.
- x. Staðværar breytur í S eru nothæfar í stofni R.  
Local variables of S are usable in the body of R.

**Svar/Answer:**

a	b	c	d	e	f	g	h	i	j	k	l
X	X	X	X	X	X	X	X	X			X

m	n	o	p	q	r	s	t	u	v	w	x
X	X	X						X			

**9. (10%)**

Svarið tveimur af eftirfarandi spurningum um ruslasöfnun.

Answer two of the following questions about garbage collection.

- a. Hver er fastayrðing gagna í ruslasöfnun sem byggist á tilvísunartalningu? Undir hvaða kringumstæðum er minnissvæði skilað?

What is the data invariant in garbage collection based on reference counting? Under what circumstances is a memory area freed?

- b. Ef forritið úthlutar mikið af skammlífum minnissvæðum, hvort er þá hraðvirkara að nota merkja og sópa eða afritunarsöfnun? Útskýrið svarið.

If the program allocates a lot of short-lived memory areas, which of the methods, mark and sweep, or stop and copy, are fastest? Explain your answer.

- c. Í sumum forritunarmálum eru vakningarfærslur í kös. Nefnið three slík forritunarmál. Útskýrið á hvaða hátt notkun lokana getur verið öflugri í slíkum forritunarmálum.

In some programming languages activation records are in the heap. Name three such programming languages. Explain in what way the use of closures can be more powerful in such programming languages.

**Svar/Answer:**

- a. Í tilvísunartalningu inniheldur sérhvert minnissvæði í kösinni teljara sem inniheldur fjölda beinna tilvísana á það minnissvæði. Minnissvæðinu er skilað þegar teljarinn verður núll.
- b. Ef mikið er notað af skammlífum minnissvæðum er hraðvirkara að nota afritssöfnun vegna þess að með þeirri aðferð er enginn tímakostnaður við að skila svæðinu. Í merkja og sópa þarf hins vegar að setja minnissvæðið í frjálsa listann, sem er vinnsla sem tekur tíma sem verður í hlutfalli við fjölda þeirra svæða sem skilað er.
- c. Í Scheme, CAML og Haskell eru vakningarfærslur í kös. Þessi forritunarmál gefa kost á að skila földuðu falli (lokun) sem gildi úr falli. Það væri ekki hægt með góðu móti ef vakningarfærslur væru á hlaða vegna þess að þá gæti svo farið að þegar kallað verður á lokunina þá sé það umhverfi (þær vakningarfærslur) ekki lengur til staðar sem þarf til að lokunin geti notað breytur í efri földunarhæðum.





**10. (10%)**

Vakningarfærsla falls í bálkmótuðu forritunarmáli eins og Scheme inniheldur sum eftirfarandi atriða. Hver? Eitt rangt svar gefur núll stig. The activation record (stack frame) of a function in a block-structured programming language such as Scheme contains some of the following. Which? One wrong answer gives zero points.

- a) Víðværar breytur sem eru aðgengilegar í fallinu. The global variables that are accessible in the function.
- b) Staðværar breytur fallsins sem kallaði á fallið. The local variables of the function that called the function.
- c) Viðföng fallsins. The arguments of the function.
- d) Aðgangshlekk (tengihlekk). An access link (static link).
- e) Stýrihlekk. A control link (dynamic link).
- f) Benda á öll föll sem hægt er að kalla á úr fallinu. Pointers to all the functions that can be called from the function.
- g) Vendivistfang. A return address.
- h) Benda á allar lifandi vakningarfærslur. Pointers to all living activation records.
- i) Alla hluti sem til eru í kerfinu. All objects that exist in the system.
- j) Vakningarfærslur allra falla sem hægt er að kalla á. The activation records of all functions that can be called.
- k) Skráakerfi tölvunnar. The file system of the computer.
- l) Nöfn allra falla sem hægt er að kalla á. The names of all functions that can be called.
- m) Staðværar breytur fallsins. The local variables of the function.

**Svar/Answer:**

a	b	c	d	e	f	g	h	i	j	k	l	m
		X	X	X		X						X

(Aukablaðsíða / Extra page)

(Aukablaðsíða / Extra page)

(Aukablaðsíða / Extra page)

(Aukablaðsíða / Extra page)