TÖL304G Final Exam
TÖL304G Lokapróf

Nafn/Name: ……..

Háskólatölvupóstfang/University Email: ……..

1. No help materials are allowed.
   Engin hjálpargögn eru leyfileg.
2. Write your answers on these pages, not in an exam book.
   Skrifið svörin á þessar síður, ekki í prófbók.
3. 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 22”.
   Ef svarið kemst ekki fyrir á tilteknu svæði má skrifa á auðar síður aftast, en þá skalt þú láta vita af því með því að skrifa tilvísun í tiltekið svæði, til dæmis “framhald á blaðsíðu 22”.
4. 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.
   Forðist að skemma eða rífa þessar síður, þær þurfa að fara gegnum skanna. Skrifið skýrt með **dökku letri** og ekki skrifa í spássíur.
5. The backs of the pages **will not be scanned** and can be used for scratch. Any answers written on the backs **will be ignored**.

Baksíður **verða ekki skannaðar** og má nota fyrir krass. **Ekki verður tekið tillit til** svara sem skrifuð eru á baksíður.

6. The exam is divided into **parts**. Answer **10** questions in total and at least **the specified required number** from each of the parts.
   Prófið skiptist í **hluta**. Svarið **10** spurningum í heild og að minnsta kosti **tilteknum lágmarksfjölda** í hverjum hluta.
7. If you answer more than **10** questions then your grade will be computed as the **average of all questions answered** unless you clearly **cross out** answers you do not want to count. You must cross out all the answer, not just part of it.
   Ef þú svarar fleiri en **10** spurningum þá verður einkunn þín reiknuð sem **meðaltal allra svara** nema þú **krossir skýrt út** svör sem þú vilt ekki að gildi. Þú verður að krossa út allt svarið, ekki aðeins hluta þess.
8. Remember that all functions need a **description** with Usage/Pre/Post or Usage/Pre/Value.
   Munið að öll föll þurfa **notkunarlýsingu** með Notkun/Fyrir/Eftir eða Notkun/Fyrir/Gildi.
9. Remember to use proper **indentation** in all program code.
   Munið að nota viðeigandi **innfellingu** í öllum forritstexta.

Part I – Block-Structure, etc.
Hluti I – Bálkmótun o.fl.

Answer at least two questions in this part – Remember to answer at least 10 questions in total
Svarið að minnsta kosti tveimur spurningum í þessum hluta – Munið að svara a.m.k. 10 spurningum í heild

1.

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 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 innihalda fallsbendi. Closures contain a function pointer.
f) Lokanir eru nauðsynlegar til að skila staðværu falli sem skilagildi falls í bálkmótuðum forritunarmálum. Closures are necessary in order to return a local function as a return value of a function in block structured programming languages.
g) Lokanir eru aðeins mögulegar ef vakningarfærslur eru í kös. Closures are only possible if activation records are in the heap.
h) Lokanir má nota til að útfæra strauma í Scheme. Closures can be used to implement streams in Scheme.
i) Lokanir innihalda stýrihlekk. Closures contain a control link (dynamic link).
j) Lokanir innihalda tengihlekk (aðgangshlekk). Closures contain an access link (static link).
k) Lokanir innihalda straum. Closures contain a stream.
l) Lokanir eru nauðsynlegar til að senda staðvær föll sem viðföng í föll í bálkmótuðum forritunarmálum. Closures are necessary in order to pass local functions as arguments to functions in block structured programming languages.

Svar/Answer:

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

2.

Vakningarfærsla falls í bálkmótuðu forritunarmáli eins og Scheme inniheldur sum eftirfarandi atriða. Hver? Tvö röng svör gefa 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? Two wrong answers give zero points.

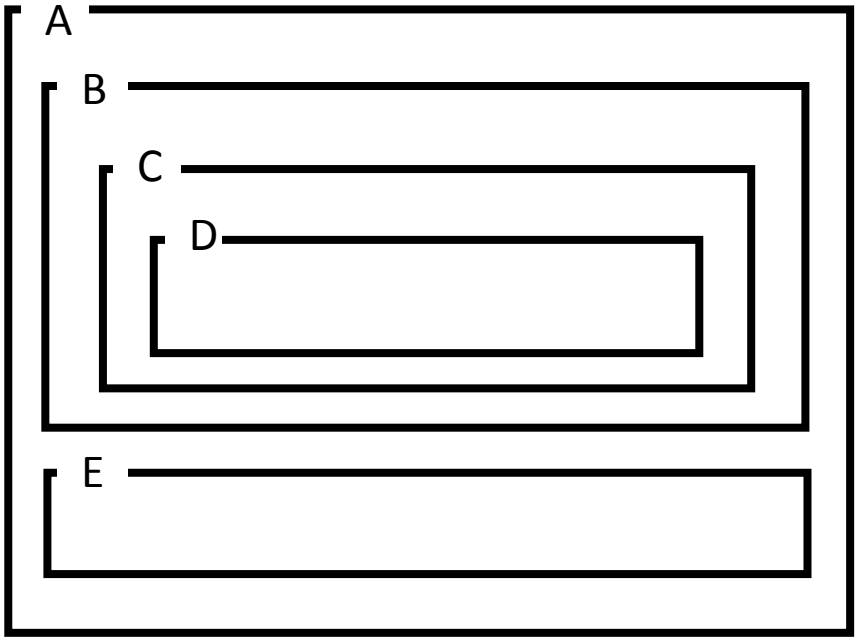
a) Staðværar breytur fallsins. The local variables of the function.
b) Bendi á vakningarfærslu fallsins sem kallaði á fallið. A pointer to the activation record of the function that called the function.
c) Bendi á vakningarfærslu fallsins sem inniheldur fallið, textalega séð, ef eitthvert er. A pointer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

Svar/Answer:

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

3.

Íhugið myndina sem sýnir földun falla A, B, C, D og E.



Samsvarandi Scheme forritstexti er einnig sýndur í tveimur jafngildum útgáfum hlið við hlið.
Consider the figure which shows nesting of functions A, B, C, D and E. The corresponding Scheme code is also shown in two equivalent versions side by side.

```
(define (A …)
  (define (B …)
    (define (C …)
      (define (D …)
        …[stofn D/body of D]
      )
      …[stofn C/body of C]
    )
    …[stofn B/body of B]
  )
  (define (E …)
    …[stofn E/body of E]
  )
  …[stofn A/body of A]
)
```

```
(define (A …)
  (define (E …)
    …[stofn E/body of E]
  )
  (define (B …)
    (define (C …)
      (define (D …)
        …[stofn D/body of D]
      )
      …[stofn C/body of C]
    )
    …[stofn B/body of B]
  )
  …[stofn A/body of A]
)
```

Fyllið út eftirfarandi töflur með því að setja krossa við sannar fullyrðingar. Eitt rangt svar gefur núll í einkunn fyrir dæmið.
Fill out the following tables by ticking the correct assertions. One wrong

answer gives zero points for the question.

Svar / Answer:

Kalla má á A úr/A may be called from:

A	B	C	D	E
x	x	x	x	x

Kalla má á B úr/B may be called from:

A	B	C	D	E
x	x	x	x	x

Kalla má á C úr/C may be called from:

A	B	C	D	E
	x	x	x	

Kalla má á D úr/D may be called from:

A	B	C	D	E
		x	x	

Kalla má á E úr/E may be called from:

A	B	C	D	E
x	x	x	x	x

Staðværar breytur í A má nota í/Local variables in A may be used in:

A	B	C	D	E
x	x	x	x	x

Staðværar breytur í B má nota í/Local variables in B may be used in:

A	B	C	D	E
	x	x	x	

Staðværar breytur í C má nota í/Local variables in C may be used in:

A	B	C	D	E
		x	x	

Staðværar breytur í D má nota í/Local variables in D may be used in:

A	B	C	D	E
			x	

Staðværar breytur í E má nota í/Local variables in E may be used in:

A	B	C	D	E
				x

4.

Eftirfarandi forritstexti er í einhverju ímynduðu forritunarmáli.
(Spyrjið ef ykkur finnst merking eða málfræði ekki augljós.)
The following program text is in some imagined programming language. (Ask if you feel the syntax or semantics is not obvious.)

```
void f(x,y)
{
  y = 2;
  print x,y;
  x = 1;
}
int i,a[10];
for( i=0 ; i!=10 ; i++ ) a[i]=i+1;
f(a[a[0]],a[0]);
print a[0], a[1], a[2], a[3];
```

Hvað skrifar þetta forrit (sex gildi í hvert skipti) ef viðföngin eru:
What does the program write (six values each time) if the parameters are:

a) Gildisviðföng / Passed by value:
2 2
1 2 3 4

b) Tilvísunarviðföng / Passed by reference:
2 2
2 1 3 4

c) Nafnviðföng / Passed by name:
3 2
2 2 1 4

Part II – List Processing, etc.
Hluti II – Listavinnsla o.fl.

Answer at least two questions in this part – Remember to answer at least 10 questions in total
Svarið að minnsta kosti tveimur spurningum í þessum hluta – Munið að svara a.m.k. 10 spurningum í heild

5.

Skrifið halaendurkvæmt fall í Scheme, CAML, Morpho eða Haskell, sem tekur lista talna $x_1, \ldots, x_n$ sem viðfang og skilar summunni $\sum_{i=1}^{n} x_i^2$. Þið munið þurfa hjálparfall og munið að skrifa réttar notkunarlýsingar. Einungis má nota einföld innbyggð föll svo sem +, *, null? car, cdr og cons, en ekki flóknari föll svo sem foldl eða map.

Write a tail-recursive function in Scheme, CAML, Morpho or Haskell, that takes as argument a list of numbers $x_1, \ldots, x_n$ and returns the sum $\sum_{i=1}^{n} x_i^2$. You will need a helper function and remember to write correct usage descriptions. You may only use simple built-in functions such as +, *, null?, car, cdr and cons, but not more complicated functions such as foldl or map.

Svar/Answer:

```
;; Notkun: (sumsq x)
;; Fyrir:  x er listi talna (x1 x2 … xN)
;; Gildi:  x1^2+x2^2+…+xN^2
(define (sumsq x)
  ;; Notkun: (help u z)
  ;; Fyrir:  u er tala,
  ;;         z er listi talna (z1 z2 … zM)
  ;; Gildi:  u+z1^2+z2^2+…+zM^2
  (define (help u z)
    (if (null? z)
        u
        (help (+ u (* (car x) (car x))) (cdr z))
    )
  )
  (help 0 x)
)
```

6.

Skrifið halaendurkæmt fall zipMapRev í Scheme, CAML, Morpho eða Haskell sem tekur tvö viðföng sem eru jafnlangir listar. Fyrra viðfangið skal vera listi einundarfalla, $f_1, \ldots, f_n$, og seinna viðfangið skal vera listi gilda $x_1, \ldots, x_n$ þannig að sérhvert x_i er löglegt viðfang í samsvarandi f_i. Fallið skal skila viðsnúnum lista gildanna sem föllin skila þegar þeim er beitt á gildin, þ.e. lista með gildunum $f_n(x_n), \ldots, f_1(x_1)$, í þeirri röð. Notið einungis einfaldar aðgerðir svo sem car, cdr, cons, null?. Í Morpho má nota lykkju, með fastarðingu lykkju.

Write a tail-recursive function zipMapRev in Scheme, CAML, Morpho or Haskell that takes two arguments that are equally long lists. The first argument shall be a list of unary functions, $f_1, \ldots, f_n$, and the second shall be a list of values $x_1, \ldots, x_n$ such that each x_i is a valid argument for the corresponding f_i. The function shall return a reversed list of the values returned by the functions when applied to the values, i.e. a list with the values $f_n(x_n), \ldots, f_1(x_1)$, in that order. Use only simple operations such as car, cdr, cons and null?. In Morpho a loop may be used, with a loop invariant.

Svar/Answer:

```
;; Notkun: (zipMapRev f x)
;; Fyrir: f er listi einundarfalla (f1 f2 … fN),
;;        x er jafn langur listi gilda (x1 x2 … xN),
;;        þar sem sérhvert xI er löglegt viðfang fyrir
;;        samsvarandi fall fI.
;; Gildi: Öfugi listinn af útkomunum
;;          ((fN xN) … (f2 x2)(f1 x1))
(define (zipMapRev f x)
  ;; Notkun: (hjalp f x z)
  ;; Fyrir: f er listi einundarfalla (f1 f2 … fN),
  ;;        x er jafn langur listi gilda (x1 x2 … xN),
  ;;        þar sem sérhvert xI er löglegt viðfang fyrir
  ;;        samsvarandi fall fI,
  ;;        z er listi (z1 z2 … zK)
  ;; Gildi: Öfugi listinn af útkomunum með z skeyttu við
  ;;          ((fN xN) … (f2 x2) (f1 x1) z1 z2 … zK)
  (define (hjalp f x z)
    (if (null? f)
        z
        (hjalp (cdr f) (cdr x) (cons ((car f) (car x)) z))
    )
  )
  (hjalp f x '())
)
```

7.

Skrifið ykkar eigin útgáfur af föllunum tveimur sem í CAML Light eru kölluð it_list og list_it. Í Haskell eru þau kölluð foldl og foldr. Þið megið skrifa þessi föll í Scheme, CAML, Morpho eða Haskell. Notið ekki lykkjur í Morpho. Kallið föllin myLeft og myRight. Þið megið nota aðra röð viðfanga en í it_list og list_it. Sjáið til þess að a.m.k. annað fallið sé halaendurkvæmt og tiltakið hvort það er. Notið aðeins einföld innbyggð föll svo sem car, cdr og null?.
Write your own versions of the two functions that in CAML Light are called it_list and list_it. In Haskell they are called foldl and foldr. You may write these functions in Scheme, CAML, Morpho or Haskell. Do not use loops in Morpho. Name the functions myLeft and myRight. You may use another ordering of argument than that used in it_list and list_it. Make sure that at least one of the functions is tail recursive and indicate which one that is. Use only simple builtin functions such as car, cdr and null?.

Halaendurkvæma fallið er/The tail recursive function is:
`foldl`

Forritstexti (með lýsingum)/Program text (with descriptions):

```
;; Notkun: (foldl f u x)
;; Fyrir:  u er gildi,
;;         x=(x1 … xN) er listi gilda,
;;         f er tvíundarfall.
;; Gildi:  (f (f … (f (f u x1) x2)…) xN)
(define (foldl f u x)
   (if (null? x)
       u
       (foldl f (f u (car x)) (cdr x))
   )
)
```

```
;; Notkun: (foldr f x u)
;; Fyrir:  u er gildi,
;;         x=(x1 … xN) er listi gilda,
;;         f er tvíundarfall.
;; Gildi:  (f x1 (f x2 (f … (f xN u)…)))
(define (foldr f x u)
   (if (null? x)
       u
       (f (car x) (foldr f (cdr x) u))
   )
)
```

8.

Skrifið fall **mapreduce** í Scheme, CAML, Morpho eða Haskell þannig að (í CAML) segðin **mapreduce op f x u** sé jafngilt segðinni **list_it op (map f x) u**. Notið aðeins einfaldar innbyggðar aðgerðir í lausninni, so sem hd, tl, :: og ==. Ekki má nota map eða list_it. Munið að list_it reiknar frá hægri til vinstri.

Write a function mapreduce in Scheme, CAML, Morpho or Haskell so that the expression **mapreduce op f x u** is equivalent to the expression **list_it op (map f x) u**. Use only simple built-in functions in the solution, such as hd, tl, :: and ==. Do not use map or list_it. Remember that list_it computes from right to left.

Svar/Answer:

```
(* Notkun: mapreduce op f x u
** Fyrir:  op er tvíundarfall af tagi 'a->'b->'b,
**         f er einundarfall af tagi 'c->'a,
**         x=[x1;…;xN] er listi gilda af tagi 'c
**         u er gildi af tagi 'b
** Gildi:  Skilagildið er sama og fyrir segðina
**         list_it op (map f x) u, þ.e.
**         (f x1)+(f x2)+…+(f xN)+u,
**         þar sem + stendur fyrir op, reiknað
**         frá hægri til vinstri.
*)
let rec mapreduce op f x u =
  if x==[] then
    u
  else
    op (f (hd x)) (mapreduce op f (tl x) u)
```

Part III – Modular Programming, etc.
Hluti III – Einingaforritun o.fl.

Answer at least two questions in this part – Remember to answer at least 10 questions in total
Svarið að minnsta kosti tveimur spurningum í þessum hluta – Munið að svara a.m.k. 10 spurningum í heild

9.

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 forgangsbiðröðina. 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:

```
{;;;
Hönnun
======

Útflutt atriði
--------------

Notkun: pq = makePQ();
Fyrir:  Ekkert
Eftir:  pq er ný tóm forgangsbiðröð

Innflutt atriði
---------------

Notkun: b = x <<== y;
Fyrir:  x og y eru gildi af þeirri gerð sem við notum
        sem lykla í forgangsbiðröðum
Eftir:  b er satt ef lykillinn x má vera á undan
        lyklinum y

Notkun: b = pq.isEmpty();
Fyrir:  pq er forgangsbiðröð
Eftir:  b er satt þá og því aðeins að pq sé tóm

Notkun: g = pq.get();
Fyrir:  pq er forgangsbiðröð, ekki tóm
Eftir:  g er gildi sem var fjarlægt úr pq.
        g hafði lykil sem mátti vera fyrir framan
        lykla allra annara gilda í pq.

Notkun: pq.add(k,g);
Fyrir:  pq er forgangsbiðröð,
        k er lykilgildi, þ.e. gildi sem er löglegt
        viðfang í innfluttu <<==.aðgerðina.
Eftir:  Búið er að bæta gildinu g í forgangsbiðröðina
        undir lyklinum k.

Smíðin er á næstu síðu.
;;;}
```

```
"pq.mmod" =
{{
makePQ =
  obj()
  {
    var list = [];
    {;;;
      Fastarðing gagna:
        Forgangsbiðröð sem inniheldur lykla/gilda-pör
        (k1,g1),…,(kN,gN), þar sem fremri lyklar í
        rununni mega vera fyrir framan aftari lykla,
        er geymd með list==[[k1$g1],…,[kN$gN]].
    ;;;}
    msg isEmpty()
        {
          …
        };
    msg put(k,g)
        {
          …
        };
    msg get()
        {
          val res = tail(head(list));
          list = tail(list);
          return res;
        };
  }
}}
;
```

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. Notendur einingar geta breytt fastayrðingu gagna einingarinnar. The users of a module can change the data invariant of the module.
b. 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.
c. Notendur einingar eiga að vita hver fastayrðing gagna er fyrir eininguna.
   The users of a module should know what the data invariant of the module is.
d. Fastayrðing gagna einingar skal halda leyndri fyrir smiðum einingarinnar.
   The data invariant of a module should not be known to the implementors of the module.
e. Fastayrðing gagna skal vera hluti af opinberu hönnunarskjali einingarinnar.
   The data invariant should be part of the public design document for the module.
f. 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.
g. Smiðir einingar geta breytt fastayrðingu gagna einingarinnar.
   The implementors of a module can change the data invariant of the module.

Svar/Answer:

a	b	c	d	e	f	g
					x	x

11.

Íhugið klasa A og B þar sem B er undirklasi A. Gerið ráð fyrir að klasi A innihaldi boð f með eftirfarandi lýsingu.

// Notkun: z = t.f(x);
// Fyrir: $0.5 \leq x \leq 1.5$
// Eftir: $\left|z - \sqrt{x}\right| < 0.01$

Gerið ráð fyrir að í klasa B sé boðið f endurskilgreint með

// Notkun: z = t.f(x);
// Fyrir: F_B
// Eftir: E_B

Consider classes A and B where B is a subclass of A. Assume that class A contains a message f with the following description.

// Usage: z = t.f(x);
// Pre: $0.5 \leq x \leq 1.5$
// Post: $\left|z - \sqrt{x}\right| < 0.01$

Assume that in class B the message f is redefined with

// Usage: z = t.f(x);
// Pre: F_B
// Post: E_B

Hverjir af eftirfarandi möguleikum fyrir F_B og E_B væru þá í lagi? Eitt rangt svar gefur núll fyrir dæmið.

Which of the following possibilities for F_B and E_B would then be acceptable? One wrong answer yields zero for the question.

Svar / Answer:

F_B	**Í lagi / Acceptable**
$0.9 \leq x \leq 1.5$	
$0.5 < x \leq 1.5$	
$0.1 \leq x \leq 5.5$	**x**
$0.2 \leq x < 1.5$	

E_B	**Í lagi / Acceptable**
$\left\|z - \sqrt{x}\right\| < 0.01$	**x**
$\left\|z - \sqrt{x}\right\| < 0.1$	
$\left\|z - \sqrt{x}\right\| < 0.001$	**x**

Part IV – Miscellaneous
Hluti IV – Ýmislegt

Answer at least one question in this part – Remember to answer at least 10 questions in total
Svarið að minnsta kosti einni spurningu í þessum hluta – Munið að svara a.m.k. 10 spurningum í heild

12.

Lýsið einum kosti sem ruslasöfnunaraðferðin merkja og sópa hefur fram yfir afritunarsöfnun og einum kosti sem afritunarsöfnun hefur fram yfir merkja og sópa.
Describe one advantage that the garbage collection method mark and sweep has over the stop and copy method and one advantage that the stop and copy method has over mark and sweep.
Svar/Answer:

Merkja og sópa leyfir stærri kös og stærra heildarminnissvæði í notkun.

Afritunarsöfnun er hraðvirkari þegar minnisúthlutun er tíð og minnissvæði skammlíf.

13.

a.

Íhugið eftirfarandi EBNF skilgreiningu fyrir mál yfir stafrófið $\{(,),a\}$. Consider the following EBNF definition for a language over the alphabet $\{(,),a\}$.

e = ‘a’ | e, ‘(‘, e, ‘)’ ;

Hverjir eftirfarandi strengja eru í málinu sem skilgreint er? Tvö röng svör gefa núll stig.

Which of the following strings are in the language defined? Two wrong answers give zero points.

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

Svar:

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

b. Hverjar af eftirfarandi reglulegu segðum skilgreina eitthvert mál yfir stafrófið $\{a,b\}$ þar sem a og b eru í jafnvægi, þ.e. strengi þannig að ef öllum $'a'$ væri breytt í $'('$ og öllum $'b'$ breytt í $')'$ þá væru svigar í jafnvægi? Tvö röng svör gefa núll stig.

Which if the following regular expressions define some language over the alphabet $\{a,b\}$ where a and b are balanced, i.e. strings such that if all $'a'$ were to be changed to $'('$ and all $'b'$ changed to $')'$ then parentheses would be balanced? Two wrong answers give zero points.

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:

1	2	3	4	5	6	7	8
	x	x		x		x	x

(auð blaðsíða/empty page)

(auð blaðsíða/empty page)

(auð blaðsíða/empty page)

(auð blaðsíða/empty page)