

HANDBÓK FYRIR AFBRIGÐI AF MICROMORPHO

Verkefni í TÖL202M Þýðendur
Snorri Agnarsson – Háskóli Íslands

Bjarki Baldursson Harksen
bbh14@hi.is

26. apríl 2022

Efnisyfirlit

1	Inngangur	2
2	Notkun og uppsetning	2
2.1	Uppsetning	2
2.2	Notkun	2
3	Málfræði	3
3.1	Lykilorð og sértákn	3
3.2	Athugasemdir	3
3.3	Forrit og setningar	3
3.4	Breytunöfn	5
3.5	Lesfastar	6
4	Merkingarfræði	9
4.1	Forrit og stofnar	9
4.2	Gildi	9
4.3	Lesfastar	9
4.3.1	Heiltölufastar	9
4.3.2	Fleytitölufastar	9
4.3.3	Strengfastar	9
4.3.4	Staffastar	9
4.3.5	Rökfastar	9
4.3.6	Tómatilvísun	10
4.4	Breytur og föll	10
4.4.1	Umdæmi	10
4.4.2	Breytuskilgreiningar	10
4.4.3	Föll og lokanir	10
4.4.4	Aðgerðir	11
4.4.5	BASIS einingin	11
4.5	Segðir	11
4.5.1	Lesfastasegð	11
4.5.2	Breytusegð	11
4.5.3	Gildisveiting	12
4.5.4	Listasegð	12
4.5.5	Kallsegð	12
4.5.6	Aðgerðasegð	12
4.5.7	Sannlegar segðir og röksegðir	13
4.5.8	Segð í sviga	13
4.5.9	Skilyrt segð	14
4.5.10	Lykkjusegð	14
4.5.11	Lokanir	14
4.5.12	Skilsegð	14
4.5.13	Samskeið segð	15

1 Inngangur

Þessi handbók gerir grein fyrir málfræði og merkingu afbrigðis af forritunarmálinu MicroMorpho, ásamt notkun á þýðanda þess. Þetta afbrigði er frábrugðið venjulegri skilgreiningu á MicroMorpho, en málin eru nógu lík til þess að ekki sé nauðsynlegt að gera greinarmun þeirra á milli. Við munum því nota MicroMorpho í meiningu afbrigðisins í framhaldi þessarar handbókar.

MicroMorpho er gildingarmál (e. imperative language) í anda C og ættaðra mála. Málfræði þess er keimlík málfræði Morpho og er MicroMorpho að miklu leyti undirmál þess. Þar að auki notar MicroMorpho Morpho sýndarvélina sem undirstöðu, en MicroMorpho þýðandinn notar hana til keyrslu Morpho vélarmáls.

2 Notkun og uppsetning

2.1 Uppsetning

Ef git er til staðar má nota eftirfarandi skeljarskipun til þess að sækja MicroMorpho þýðandann `mm-compiler`.

```
$ git clone https://github.com/mrbjarksen/mm-compiler/
```

Þetta ætti að búa til möppuna `mm-compiler` með frumkóða þýðandans. Einnig má sækja þessa möppu beint frá slóðinni sem gefin er í skipuninni.

Áður en hægt er að setja upp þýðandann þarf að hafa aðgang að Haskell þýðanda og smíðatólinu Cabal. Nægja ætti að setja upp GHCup (sjá nánar hér) Að þessu loknu þarf aðeins að keyra eftirfarandi skipun innan möppunnar:

```
$ cabal install
```

Þetta býr til keyranlega útgáfu af þýðandanum og setur hana í `$HOME/.cabal/bin/` á UNIX vélum, en í `%APPDATA%\cabal\bin` á Windows. Bæta má viðeigandi möppu við umhverfisbreytuna `PATH` ef keyra vill þýðandann frá skipanalínu.

2.2 Notkun

Þýðing og keyrsla MicroMorpho forrita með þýðandanum `mm-compiler` krefst þess að Java sé sett upp (sjá heimasíðu Java). Hins vegar má nota þýðandann til lesgreiningar, þáttunar og þýðingar í Morpho vélarmál jafnvel þótt Java sé ekki til staðar.

Til þess að sýna notkun þýðandans skulum við skrifa einfalt MicroMorpho forrit. Við byrjum á því að búa til skrána `hello.mm` eins og að neðan.

```
hello.mm
writeln("Hello, world!");
```

Nú þýðum við forritið:

```
$ mm-compiler hello.mm
```

Þetta býr til skrána `hello.mexe` í sömu möppu. Nú þarf aðeins að keyra forritið með eftirfarandi skipun:

```
$ mm-compiler run hello.mexe  
Hello, world!
```

Til þess að sjá fleiri leiðir sem nota má þýðandann ætti að lesa hjálpartexta forritsins. Hann má sjá með stikanum `--help`.

```
$ mm-compiler --help
```

3 Málfræði

3.1 Lykilorð og sértákn

Eftirfarandi orð eru lykilorð í MicroMorpho:

`else, elsif, false, fun, go, if, null, return, true, var, while`

Þessi orð er því ekki hægt að nota sem nöfn fyrir föll eða breytur.

Auk þessara lykilorða hafa eftirfarandi tákn sérstaka merkingu:

`(,), {, }, [,], ,, ;, =`

Þessi sértákn verka jafnframt sem afmarkarar á milli annarra frumeininga málsins. Sama á við um hvers kyns hvíttákn (e. `whitespace`).

3.2 Athugasemdir

Í MicroMorpho eru tvær gerðir athugasemda, annars vegar línuathugasemdir og hins vegar bálkathugasemdir. Texti innan í athugasemd er hunsaður.

Línuathugasemd byrjar með `;;;` og nær að enda línunnar. Bálkathugasemd byrjar með `{;;;` og endar með `;;;}`. Bálkathugasemd má vera innan í annarri bálkathugasemd, þ.e.a.s. ef bálkathugasemd hefst innan í annarri bálkathugasemd, þá þarf að loka þeirri innri áður en lokað er þeirri ytri. Sérhverri bálkathugasemd sem ólokið er við skráarlök er lokað sjálfkrafa.

3.3 Forrit og setningar

MicroMorpho forrit er listi af setningum sem afmarkaðar eru með einni eða fleiri semikommum. Ekki er nauðsynlegt að ljúka síðustu setningu forrits með semikomu, en það er leyfilegt.

```

 $\langle program \rangle ::= \langle stmts \rangle$ 

 $\langle stmts \rangle ::= \langle stmt \rangle ; \langle scs \rangle \langle stmts \rangle$ 
               |  $\langle stmt \rangle \langle scs \rangle$ 

 $\langle stmt \rangle ::= \langle fundecl \rangle$ 
               |  $\langle vardecl \rangle$ 
               |  $\langle expr \rangle$ 

 $\langle scs \rangle ::= ; \langle scs \rangle$ 
               |  $\epsilon$ 

```

Mállýsing 1: Forrit

Setningar forrits skiptast í skilgreiningar falla, skilgreiningar breyta og segðir. Eftirfarandi eru mállýsingar hverra þessara setninga.

Skilgreining falla og aðgerða nota sama rithátt; á þeim er enginn formlegur greinarmunur. Þessi ritháttur er auðkenndur með lykilorðinu **fun**.

```

 $\langle fundecl \rangle ::= \mathbf{fun} \langle name \rangle ( \langle names \rangle ) \langle body \rangle$ 
               |  $\mathbf{fun} \langle opname \rangle ( \langle names \rangle ) \langle body \rangle$ 

 $\langle names \rangle ::= \langle names1 \rangle$ 
               |  $\epsilon$ 

 $\langle names1 \rangle ::= \langle name \rangle , \langle names1 \rangle$ 
               |  $\langle name \rangle$ 

 $\langle body \rangle ::= \{ \langle stmts \rangle \}$ 

```

Mállýsing 2: Fallsskilgreining

Breytuskilgreiningar nota lykilorðið **var**. Skilgreina má margar breytur í einni breytuskilgreiningu. Auk þess má velja að frumstilla breytu um leið og hún er skilgreind.

```

 $\langle vardecl \rangle ::= \mathbf{var} \langle vardecls \rangle$ 

 $\langle vardecls \rangle ::= \langle declorinit \rangle , \langle vardecls \rangle$ 
                  |  $\langle declorinit \rangle$ 

 $\langle declorinit \rangle ::= \langle name \rangle$ 
                  |  $\langle name \rangle = \langle expr \rangle$ 

```

Mállýsing 3: Breytuskilgreining

Stór hluti MicroMorpho heyrir undir segðir. Nánari útskýringu á hinum ýmsu segðum málsins má finna í 4.5.

```

⟨expr⟩ ::= return ⟨expr⟩
        | ⟨name⟩ = ⟨expr⟩
        | go ⟨body⟩
        | ⟨expr⟩ ⟨opname⟩ ⟨expr⟩
        | ⟨opname⟩ ⟨expr⟩
        | ⟨name⟩
        | ⟨expr⟩ ( ⟨exprs⟩ )
        | ⟨literal⟩
        | [ ⟨exprs⟩ ]
        | ( ⟨expr⟩ )
        | if ( ⟨expr⟩ ) ⟨body⟩ ⟨elsifs⟩ ⟨else⟩
        | while ( ⟨expr⟩ ) ⟨body⟩
        | fun ( ⟨names⟩ ) ⟨body⟩

⟨exprs⟩ ::= ⟨exprs1⟩
        | ε

⟨exprs1⟩ ::= ⟨expr⟩ , ⟨exprs1⟩
        | ⟨expr⟩

⟨elsifs⟩ ::= elsif ( ⟨expr⟩ ) ⟨body⟩ ⟨elsifs⟩
        | ε

⟨else⟩ ::= else ⟨body⟩
        | ε

```

Mállýsing 4: Segðir

3.4 Breytunöfn

Nöfn á breytum (og föllum) mega innihalda há- eða lágstafi úr enska stafrófinu, tölustafi og undirstrik. Nöfn mega hins vegar ekki hafa tölustaf sem fyrsta staf.

```

⟨name⟩      ::= ⟨name-first⟩ ⟨name-rest⟩

⟨name-first⟩ ::= ⟨ascii-alpha⟩
                | -

⟨name-rest⟩  ::= ⟨ascii-alpha⟩
                | ⟨digit⟩
                | -

⟨ascii-alpha⟩ ::= a | b | c | d | e | f | g | h | i | j | k | l | m
                | n | o | p | q | r | s | t | u | v | w | x | y | z
                | A | B | C | D | E | F | G | H | I | J | K | L | M
                | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

```

Mállýsing 5: Breytunöfn

Nöfn einundar- og tvíundaraðgerða uppfylla mállýsinguna hér að neðan.

```

⟨opname⟩ ::= ⟨opelem⟩ ⟨opname⟩
           |  ⟨opelem⟩

⟨opelem⟩ ::= * | / | % | + | - | < | > | ! | = | & | | | : | ? | ~ | ^

```

Mállýsing 6: Aðgerðanöfn

3.5 Lesfastar

Lesfastar í MicroMorpho eru af sex gerðum: heiltölufastar, fleytitölufastar, strengfastar, staffastar, rökfastar og tómatilvísun. Rökfastar skiptast í **true** og **false**. Tómatilvísun er táknueð með **null**.

Samofin mállýsing allra lesfasta er eftirfarandi.

```

⟨literal⟩ ::= ⟨int⟩
              | ⟨float⟩
              | ⟨string⟩
              | ⟨char⟩
              | true
              | false
              | null

```

Mállýsing 7: Lesfastar

Mállýsingar fyrir sértæka lesfasta má nú sjá að neðan.

Heiltölufasti er einfaldlega runa af tölustöfum, sem hugsanlega hefur formerki fremst.

```

⟨int⟩      ::= ⟨sign⟩ ⟨digits⟩

⟨sign⟩     ::= +
              | -
              | ε

⟨digits⟩   ::= ⟨digit⟩ ⟨digits⟩
              | ⟨digit⟩

⟨digit⟩    ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

Mállýsing 8: Heiltölufastar

Fleytitölufasti hefur runu af tölustöfum með punkt í miðjunni, þannig að a.m.k. einn tölustafur komi fyrir hvoru megin við hann. Auk þess er hugsanlega formerki fremst í fastanum. Því næst fylgir valkvæður veldisvísir. Þessi veldisvísir byrjar annaðhvort á e eða á E og þeim staf fylgir runa af tölustöfum, hugsanlega með formerki fyrir framan.

```

⟨float⟩     ::= ⟨sign⟩ ⟨digits⟩ . ⟨digits⟩ ⟨exponent⟩

⟨exponent⟩ ::= e ⟨sign⟩ ⟨digits⟩
              | E ⟨sign⟩ ⟨digits⟩
              | ε

```

Mállýsing 9: Fleytitölufastar

Strengfasti er runa af leyfilegum stöfum, afmörkuðum með tvöföldum gæsalöppum hvoru megin. Leyfilegir stafir í streng eru allir stafir sem kóðun skráarinnar styður, að fráteknnum tvöföldum gæsalöppum (") og bakstriki (\). Til þess að tilgreina þá stafi í streng þarf að hafa bakstrik fyrir framan þá. Aðrar lausnarunur sem nota bakstriksstafinn til að tilgreina ákveðin tákn má sjá í mállýsingunni að neðan.


```

⟨string⟩ ::= " ⟨strelems⟩ "

⟨strelems⟩ ::= ⟨strelem⟩ ⟨strelems⟩
               | ε

⟨strelem⟩ ::= ⟨any-but-double-quote-and-backslash⟩
               | ⟨escaped⟩

⟨escaped⟩ ::= \ \
               | \ "
               | \ '
               | \ n
               | \ r
               | \ t
               | \ b
               | \ f
               | \ ⟨octal⟩
               | \ ⟨octal⟩ ⟨octal⟩
               | \ ⟨quat⟩ ⟨octal⟩ ⟨octal⟩

⟨octal⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

⟨quat⟩ ::= 0 | 1 | 2 | 3

```

Mállýsing 10: Strengfastar

Staffastar eru svipaðir og strengfastar, nema þeir innihalda aðeins einn staf og eru afmarkaðir með einföldum gæsalöppum fremur en tvöföldum. Leyfilegir stafir í staf-fasta innihalda tvöfaldar gæsalappir, ólíkt strengfasta, en einfaldar gæsalappir eru ekki leyfilegar. Nota má bakstrik til þess að sneiða hjá þessu.

```

⟨char⟩ ::= ' ⟨charelem⟩ '

⟨charelem⟩ ::= ⟨any-but-single-quote-and-backslash⟩
               | ⟨escaped⟩

```

Mállýsing 11: Staffastar

4 Merkingarfræði

4.1 Forrit og stofnar

Eins og nefnt var í 3.3 er MicroMorpho forrit listi af setningum. Keyrsla MicroMorpho forrits felst í því að reikna úr setningum þess. Setningarnar eru reiknaðar í sömu röð og þær koma fyrir í forritstextanum.

Sumar setningar og segðir innihalda stofn. Stofn er einnig listi setninga og er útreiknun þeirra í samræmi við útreiknun setninga í forriti.

4.2 Gildi

Öll gildi í MicroMorpho eru tilvísanir á Java hluti.

4.3 Lesfastar

Lesfastar eru af sex gerðum: heiltölufastar, fleytitölufastar, strengfastar, staffastar, rökfastar og tómatilvísun.

4.3.1 Heiltölufastar

Heiltölufastar mega innihalda eins marga tölustafi og vera skal. Gildi heiltölufasta er viðeigandi Java hlutur – `Integer`, `Long` eða `BigInteger` – með tilsvarende gildi. Valið á klasanum fer eftir stærð fastans.

4.3.2 Fleytitölufastar

Fleytitölufastar eru ávallt af tvöfaldri nákvæmni. Gildi þeirra er tilsvarende Java `Double` hlutur. Ef fleytitölufasti er skrifaður með meiri nákvæmni en hluturinn getur geymt er talan námunduð.

4.3.3 Strengfastar

Gildi strengfasta er tilsvarende Java `String` hlutur.

4.3.4 Staffastar

Gildi staffasta er tilsvarende Java `Character` hlutur.

4.3.5 Rökfastar

MicroMorpho hefur tvo rökfasta: `true` og `false`. Gildi þeirra er tilsvarende Java `Boolean` hlutur.

4.3.6 Tómatilvísun

MicroMorpho inniheldur tómatilvísun á forminu `null`. Þessi tómatilvísun hefur sama gildi og tómatilvísun Java.

4.4 Breytur og föll

Allar breytur í MicroMorpho eru staðværar og eru bundar við umdæmi sitt. Breytur eru aðeins sjáanlegar í þeim setningum sem koma á eftir skilgreiningu þeirra.

4.4.1 Umdæmi

Sérhvert MicroMorpho forrit hefur ysta umdæmi. Önnur umdæmi samsvara stofnum fallsskilgreininga eða `go-/if-/while-/lokanasegða`. Breytur úr ytri umdæmum eru þar sjáanlegar, en breytur sem eru skilgreindar innan slíks stofns falla úr umdæmi við lokun hans.

4.4.2 Breytuskilgreiningar

Breytuskilgreining er setning. Skilgreina má margar breytur í einni breytuskilgreiningu, og velja má að upphafstillja hverja þeirra með gildi segðar. Ef breyta er ekki upphafsstillt, þá er henni gefið gildið `null`.

Ef margar breytur eru skilgreindar í einni breytuskilgreiningu, þá eru þær skilgreindar í sömu röð og þær koma fyrir í setningunni. Þannig má nota gildi breyta sem skilgreindar hafa verið fyrr í setningunni í upphafsstillingarsegð. Ekki má nota breytu í eigin upphafsstillingarsegð.

Engar tvær breytur með sama umdæmi mega hafa sama nafn, en ef breyta er skilgreind með sama nafn og breyta í ytri umdæmi, þá vísar sérhver notkun breytunnar í umdæminu á innri skilgreiningu hennar. Þegar umdæmið lokar tekur hin gamla skilgreining við á ný.

4.4.3 Föll og lokanir

Sérhvert fall í MicroMorpho er lokun. Því má geyma fall í breytu, skila falli úr falli og nota fall sem viðfang í fall. Auk þess eru öll föll sem skilgreind eru í MicroMorpho forriti breytur. Allar reglur um breytur gilda því einnig um föll.

Þegar fall er skilgreint er fyrst athugað hvort breyta með sama nafn er skilgreind í sama umdæmi. Ef svo er er þessari breytu gefið gildið sem tilvarandi lokunarsegð hefur. Annars er breytan skilgreind og upphafsstillt með lokuninni.

Eftirfarandi tveir rithættir eru þar með jafngildir:

```
fun f() { ... };
```

```
var f;  
f = fun () { ... };
```

Hér er setningin `var f` aðeins til staðar ef ekki er búið að skilgreina `f` í sama umdæmi. Föll mega vera endurkvæm. Athuga ætti hins vegar að þar sem ekki má nota breytu í eigin upphafsstillingarsegð er ekki hægt að skilgreina endurkvæmt fall með eftirfarandi rithætti:

```
var f = fun () { ... }
```

4.4.4 Aðgerðir

MicroMorpho styður skilgreiningu einundar- og tvíundaraðgerða. Aðgerðir eru einfaldlega föll sem taka annaðhvort eitt eða tvö viðföng. Athuga ætti hins vegar að aðgerðanöfn og breytunöfn eru ekki víxlanleg, og því má ekki nota aðgerðir á sama hátt og nota má föll og breytur.

Aðgerðir eru skilgreindar með sama rithætti og föll. Ekki má skilgreina aðgerð sem hefur annan fjölda stika en einn eða tveir. Aðgerð sem skilgreind er með einum stika má nota sem einundaraðgerð en aðgerð með tveimur stikum má nota sem tvíundaraðgerð. Rökaðgerðirnar `||`, `&&` og `!` eru meðhöndlaðar á sérstakan hátt (sjá 4.5.6 og 4.5.7). Það er því ekki leyfilegt að endurskilgreina þessar aðgerðir.

4.4.5 BASIS einingin

MicroMorpho flytur inn BASIS einingu Morpho, sem inniheldur mörg grundvallarföll og aðgerðir. Þessi föll má aðeins nota í beinu fallskalli eða aðgerðarbeitingu, en föllin eru ekki sjáanleg sem breytur innan MicroMorpho forrits. Þessum föllum er ekki lýst í þessari handbók.

4.5 Segðir

Líkt og í Morpho er lögð meiri áhersla á segðir en á setningar. Einu merkingarlega sjálfstæðu fyrirbærin í MicroMorpho sem ekki eru segðir eru skilgreiningar falla og breyta. Þetta er ólíkt virkni margra C-ættaðra mála sem Morpho og MicroMorpho erfa frá.

4.5.1 Lesfastasegð

Lesfastar heyra undir segðir. Gildi lesfastasegðar er sama og gildi lesfastans (sjá 4.3).

4.5.2 Breytusegð

Nota má nafn á sjáanlegri breytu sem segð. Segðin hefur sama gildi og breytan. Breytunafn sem ekki samsvarar sjáanlegri breytu er hins vegar ekki lögleg segð.

4.5.3 Gildisveiting

Gefa má sjáanlegri breytu gildi sem er útkoma úr annarri segð. Þetta er gert með = tákninu. Ekki má gefa óskilgreindri breytu gildi á þennan hátt.

Gildisveiting er segð sem hefur sama gildi og segðin sem er notuð til þess að gefa breytu gildi. Þetta má nota til þess að gefa mörgum breytum sama gildið, til dæmis ef `a` og `b` eru skilgreindar breytur þá má gefa þeim báðum gildið 2 með segðini `a = b = 2`.

4.5.4 Listasegð

Búa má til listasegð með því að setja núll eða fleiri segðir á milli hornklofa (þ.e. táknanna `[` og `]`) afmarkaðar með kommu. Gildi segðarinnar er þá listinn sem hefur sem haus gildi fyrstu segðarinnar á milli hornklofanna (ef hún er til staðar). Tómi listinn er skrifaður `[]` og hefur sama gildi og tómatilvísunin `null`.

Þessi ritháttur er aðeins til þæginda, enda má fá sömu virkni með aðgerðinni `:` úr BASIS einingunni. Eftirfarandi segðir eru þannig jafngildar:

<code>[1,2,3]</code>	<code>1:2:3:[]</code>	<code>1:2:3:null</code>
----------------------	-----------------------	-------------------------

4.5.5 Kallsegð

Falls kall er segð sem hefur skilagildi kallsins sem gildi. Kallað er á fall með því að setja viðföng þess í sviga á eftir því, afmörkuð með kommu. Til dæmis má kalla á fall `f` sem tekur tvö viðföng með segðinni `f(1,2)`.

Föll nota ávallt gildisviðföng. Fallið sem kallað er á má vera lokun sem er útkoma úr annarri segð. Segðin að neðan er þannig lögleg og skilar gildinu 2 (ef `+` hefur ekki verið endurskilgreind).

```
(fun (x) { x + 1 })(1)
```

Ef reynt er að kalla á fall sem ekki er sjáanleg breyta, þá er leitað að viðeigandi falli í BASIS einingu Morpho.

4.5.6 Aðgerðasegð

Beiting einundar- og tvíundaraðgerðar er segð sem hefur skilagildi aðgerðinnar sem gildi. Ekki má kalla á aðgerðir á sama hátt og á föll (þ.e. með því að skeyta aftan við þau viðföngum innan sviga). Kall á einundaraðgerð hefur aðgerðarnafnið fyrir framan viðfang sitt en kall á tvíundaraðgerð hefur það á milli viðfanganna (með fyrra viðfangið til vinstri en það seinna til hægri).

Ef margar aðgerðir eru notaðar hlið við hlið og engir svigar eru notaðir til að ákvarða röðun aðgerðanna (sjá 4.5.8), þá er beitt forgangi og tengni aðgerðanna til þess að ákvarða hvaða segðir eru reiknaðar fyrst. Einundaraðgerðir hafa hæstan forgang. Forgangur og tengni tvíundaraðgerða fer eftir fyrsta staf aðgerðarinnar. Eftirfarandi tafla

sýnir forgangstölu og tengni tvíundaraðgerða eftir fyrsta staf aðgerðarinnar. Hærri forgangstala samsvarar hærri forgangi.

Fyrsti stafur	Forgangstala	Tengni
<code>*</code> , <code>/</code> eða <code>%</code>	7	Vinstri
<code>+</code> eða <code>-</code>	6	Vinstri
<code><</code> , <code>></code> , <code>!</code> , eða <code>=</code>	5	Vinstri
<code>&</code>	4	Vinstri
<code> </code>	3	Vinstri
<code>:</code>	2	Hægri
<code>?</code> , <code>~</code> eða <code>^</code>	1	Vinstri

Tafla 1: Forgangur og tengni tvíundaraðgerða

Athuga ætti hins vegar að sérstök meðhöndlun er á einundaraðgerðinni `!` og tvíundaraðgerðunum `&&` og `||`. Virkni þeirra er lýst í 4.5.7. Þessar aðgerðir hafa allar lægri forgang en aðrar aðgerðir. Innbyrðis forgangs röðun þeirra er slík að `!` hefur meiri forgang en `&&` og `&&` hefur meiri forgang en `||`. Tvíundaraðgerðirnar `&&` og `||` eru tengnar til vinstri.

4.5.7 Sannlegar segðir og röksegðir

Í MicroMorpho er sérhver segð annað hvort sögð vera sannleg eða ósannleg. Einu segðirnar sem eru ósannlegar eru lesfastasegðirnar `false` og `null`. Segðir sem nota sannleika eru skilyrtar segðir (sjá 4.5.9), lykkjusegðir (sjá 4.5.10) og röksegðir.

Röksegðir eru aðgerðarsegðir sem nota aðgerðirnar `!`, `&&` eða `||`.

- Aðgerðin `!` er nefnd neitunar-aðgerðin. Kallið `!a` skilar `true` ef `a` er ósannleg segð, annars skilar það `false`.
- Aðgerðin `&&` er nefnd og-aðgerðin. Kallið `a && b` skilar gildi segðarinnar `a` ef `a` er ósannleg segð, annars skilar það gildi segðarinnar `b`. Ef `a` er ósannleg segð er ekki reiknað úr segðinni `b`.
- Aðgerðin `||` er nefnd eða-aðgerðin. Kallið `a || b` skilar gildi segðarinnar `a` ef `a` er sannleg segð, annars skilar það gildi segðarinnar `b`. Ef `a` er sannleg segð er ekki reiknað úr segðinni `b`.

4.5.8 Segð í sviga

Ef segð er sett í sviga fæst ný segð sem hefur sama gildi og segðin innan í sviganum. Þetta er notað til þess að auka lesanleika forrits og til þess að breyta röð aðgerða.

4.5.9 Skilyrt segð

MicroMorpho styður skilyrtar segðir með lykilorðunum `if`, `elsif` og `else`. Skilyrt segð samanstendur af `if`-hluta, núll eða fleiri `elsif`-hlutum og hugsanlegum `else`-hluta. Hver hluti hefur sinn stofn og hver hluti sem ekki er `else`-hluti hefur sitt skilyrði. Dæmigerð skilyrt segð er eftirfarandi.

```
if ( $c_0$ ) { ... }  
elsif ( $c_1$ ) { ... }  
elsif ( $c_2$ ) { ... }  
:  
elsif ( $c_n$ ) { ... }  
else { ... }
```

Hér er c_0 skilyrði `if`-hlutans og c_1, \dots, c_n skilyrði `elsif`-hlutanna. Þegar reiknað er úr þessari segð er reiknað úr segðunum c_0, \dots, c_n , koll af kolli, þangað til fundin er sannleg segð. Tilsvarandi stofn er þá keyrður og hætt er að reikna skilyrðin. Ef ekkert þessara skilyrða er sannlegt þá er stofn `else`-hlutans keyrður.

4.5.10 Lykkjusegð

Lykkjusegðir í MicroMorpho einkennast af lykilorðinu `while`. Lykkjan hefur tilsvarandi skilyrði og stofn. Dæmigerð lykkjusegð er eftirfarandi.

```
while ( $c$ ) { ... }
```

Hér er c skilyrði lykkjusegðarinnar. Þegar reiknað er úr þessari segð er fyrst reiknað úr segðinni c . Ef hún er sannleg þá er stofn segðarinnar keyrður. Þetta er síðan endurtekið þangað til c er ekki sannleg segð.

4.5.11 Lokanir

Skilgreining lokunar er segð, þótt skilgreining falls sé það ekki. Gildi lokunarsegðar er lokunin sjálf. Rithættur lokunarsegðar og fallsskilgreiningar eru eins að því fráteknu að lokunarsegð hefur ekki breytunafn á eftir `fun` lykilorðinu.

4.5.12 Skilsegð

Sú aðgerð að skila gildi úr fallskalli er segð. Segðin notar `return` lykilorðið til þess að merkja gildið sem skila ætti. Fordast ætti að nota skilsegð þar sem útkoma segðarinnar skiptir máli, enda er þá hætt keyrslu fallsstofnsins.

4.5.13 Samskeið segð

MicroMorpho styður samskeiða forritun með notkun `go` lykilorðsins. Þegar `go` er sett fyrir framan stofn, þá er búinn til þráður í nýrri verkeiningu á vélinni og stofninn keyrður á þessum þræði.