

Handbók fyrir Mimo

Ólafur Óskar Egilsson

Háskóli Íslands

Tölvunarfræðideild

ooe3@hi.is | olioskar@gmail.com

23. apríl 2012

Útdráttur

Þessi handbók fylgir Mimo þýðandanum og fjallar um málið sem að hann þýðir, því er lýst með málritum og merking segða er útskýrð. Markmiðið með þessari handbók er að lesandi skilji málfræði Mimo, geti útfært einföld forrit skrifuð í málinu, þýtt þau með Mimo þýðandanum og að lokum keyrt í sýndarvél Morpho. Gert er ráð fyrir því að lesandi kunni skil á og geti túlkað málrit ásamt því að geta lesið og skilið einföld forrit óháð því máli sem þau eru skrifuð í. Að þessu gefnu ætti handbókin að nýtast lesandanum vel til skilnings á virkni, málfræði og merkingu forritunarmálsins Mimo.

Efnisyfirlit

1	Inngangur	2
2	Notkun og uppsetning	3
2.1	Uppsetning	3
2.2	Þýðing	4
3	Málfræði	5
3.1	Frumeiningar málsins	5
3.1.1	Athugasemdir	5
3.1.2	Lykilorð	5
3.1.3	Aðrar einingar málsins	5
3.2	Mállýsing	6
3.2.1	Listi yfir málrit	6
3.2.2	Málrit	7

4	Merking málsins	14
4.1	Mimo forrit	14
4.1.1	Einingar	14
4.1.2	Main() fallið	14
4.2	Gildi	14
4.3	Breytur	15
4.4	Merking segða	15
4.4.1	Heiltölusegð	15
4.4.2	Fleytitölusegð	16
4.4.3	Stafsegð	16
4.4.4	Strengsegð	17
4.4.5	return-segð	17
4.4.6	Kallsegð	17
4.4.7	Aðgerðir	18
	Forgangur	19
4.4.8	Röksegðir	19
	og-segð	19
	eða-segð	19
	ekki-segð	19
4.4.9	if-segð	19
4.4.10	while-segð	20

1 Inngangur

Mimo stendur fyrir Mini Morpho og er forritunarmál sem byggir á Morpho málinu (Morpho má finna á vefnum á slóðinni: morpho.cs.hi.is). Setningarfræði (e. syntax) málsins er einföld og ekki er hægt að segja að Mimo sé fullbúið forritunarmál. Til dæmis þá býður málið ekki upp á "for"-lykkjur heldur einungis einfaldar "while"-lykkjur. Það styður ekki hlutbundna forritun né földun. Engu að síður er hægt að útfæra bæði einföld og flókin forrit í málinu. Og þar sem setningafræðin er ekki flókin er auðvelt fyrir forritara að ná tökum á málinu. Forrit 1.1 sýnir dæmi um "Halló Heimur"forrit skrifað í Mimo og forrit 1.2 sýnir sama forrit þýtt yfir á vélamál Morpho.

Mimo þýðandinn þýðir forrit skrifað í Mimo yfir á Morpho vélamál. Síðan er vélamálsskráin þýdd af Morpho þýðandanum og úr verður keyrsluhæf skrá fyrir Morpho sýndarvélina. Það að þýða yfir í Morpho hefur marga kosti í för með sér. Til dæmis þá nýttast þýddar Mimo einingar í forritum skrifuðum í Morpho. Einnig, þá hafa Mimo forrit beinan aðgang að föllum úr BASIS einingunni - sem er grunneining í Morpho og inniheldur einundar- og tvíundaraðgerðir ásamt

föllum eins og `writeln()`.

Þetta skjal lýsir Mimo málinu og setningarfræði þess. Málinu er lýst með málritum í kafla 3.2.2 og er reiknað með að lesandi kunni að túlka slík rit. Lista yfir málrit má finna á blaðsíðu 6. Kafli 4.4 fjallar um merkingu málsins og er öllum segðum í málinu gefin skil. Vísað er í viðeigandi málrit fyrir flestar segðir en einnig eru birt einföld dæmi hvernig þær eru útfærðar í kóða. Kafli 2 fjallar um uppsetningu á Mimo þýðandanum ásamt útfærslu og þýðingu á einföldu forriti.

```
main() {  
  writeln("Halló Heimur");  
}
```

Forrit 1.1: Halló Heimur forrit í Mimo

```
"test.mexe" = main in  
!{{  
#"main[f0]" = [  
  (MakeVal "Halló Heimur")  
  (StoreArgAcc -1 0)  
  (Call #"writeln[f1]" -1)  
  (Return)  
];  
}}*BASIS;
```

Forrit 1.2: Halló Heimur forrit á Morpho vélamáli

2 Notkun og uppsetning

2.1 Uppsetning

Til að þýða og keyra Mimo forrit á linux vél þarf Java að vera uppsett á vélinni. Mimo þýðandinn þýðir yfir á Morpho vélamál og til þess að þýða það og keyra þarf Morpho að vera til staðar. Morpho.jar fylgir með, svo notandi þarf ekki að sækja það eða setja upp sérstaklega. Þar sem Mimo þýðandinn er keyrsluhæf Java skrá þá ætti ekki að vera mikið vandamál að keyra hann á Windows vélum, sé Java til staðar. Mimo hefur hinsvegar einungis verið prufað á Linux vél - svo ekki er hægt að tryggja það að þýðandinn virki á Windows vélum

2.2 Þýðing

Hægt er að fara tvær leiðir til að þýða Mimo forrit. Sé maður á Linux vél er hægt að keyra meðfylgjandi skeljarþulu *mimo* með það forrit sem þýða skal sem viðfang. Til dæmis...

```
./mimo forrit.mimo
```

Þá verða til tvær skrár önnur með endinguna *.masm* og hin með endinguna *.mexe*. Sú fyrrgreinda er Mimo forritið þýtt yfir á Morpho vélamál, sú síðari er keyrsluhæf Morpho skrá. Meðfylgjandi er einnig skeljarþula sem keyrir Morpho með viðfangi, svo eftirfarandi skipun myndi keyra forritið...

```
./morpho forrit
```

Hin leiðin er að keyra eftirfarandi runur af skipunum - Hægt er að velja hana virki skeljaþulurnar *mimo* og *morpho* ekki af einhverjum ástæðum eða sé notandi að keyra pakkann á Windows vél.

```
1: java MimoParser forrit.mimo > forrit.masm
2: java -jar morpho.jar -c forrit.masm
3: java -jar morpho.jar forrit
```

Lína 1 þýðir *forrit* yfir á Morpho vélamál

Lína 2 tekur vélamálsskránna og þýðir yfir á keyrsluhæfa Morpho skrá

Lína 3 keyrir síðan forritið

Vítt og dreyft í þessu skjali eru lítil forrit eða forritsbútar sem ættu að gefa skýra mynd af hvernig Mimo forrit eru uppbyggð. Lesandi getur byrjað á að afrita Halló Heimur forrit 1.1 á blaðsíðu 3 í hreina texta skrá og þýtt og keyrt til að ganga úr skugga um að allt sé rétt sett upp. Einnig þá fylgir með pakkanum skjalið *test:mimo* og getur lesandi stúderað kóðann í því til að kynna sér hvernig Mimo forrit líta út.

3 Málfræði

3.1 Frumeiningar málsins

3.1.1 Athugasemdir

```
f() {  
;;; Athugasemd  
var x;  
x = x+1; ;;; Athugasemd  
x = x+x;  
{;;; athugasemd sem  
spannar 2 línur ;;;}  
x = x+x;  
}
```

Forrit 3.1: Dæmi um athugasemdir

Mimo býður uppá 2 gerðir af athugasemdum. Athugasemdir sem spanna eina línu, táknuð með ';;;' og athugasemdir sem spanna margar línur, táknuð með '{;;;' í upphafi athugasemdar og ';;;}' í lok. Þessar athugasemdir eru af sömu gerð og í forritunarmálinu Morpho og lúta sömu lögmálum. Forrit 3.1 sýnir dæmi um athugasemdir í Mimo kóða.

3.1.2 Lykilorð

Eftirfarandi lykilorð eru frátekin fyrir Mimo:

`null, true, false, if, elseif, else, while, var, return`

3.1.3 Aðrar einingar málsins

Vert er að nefna lesfastar (e. literals) `<int>` `<float>` `<string>` og `<char>` og frumeiningarnar `<op>` og `<name>` sem koma fram í mállýsingu.

- `<int>`, heiltölur lýst í málriti 3.23 á blaðsíðu 11
- `<float>`, fleytitölur lýst í málriti 3.25 á blaðsíðu 12
- `<string>`, strengir lýst í málriti 3.26 á blaðsíðu 12
- `<char>`, stafir lýst í málriti 3.27 á blaðsíðu 13

- <op>, aðgerðir lýst í málriti 3.13 á blaðsíðu 9
- <name>, nöfn lýst í málriti 3.5 á blaðsíðu 7

3.2 Mállýsing

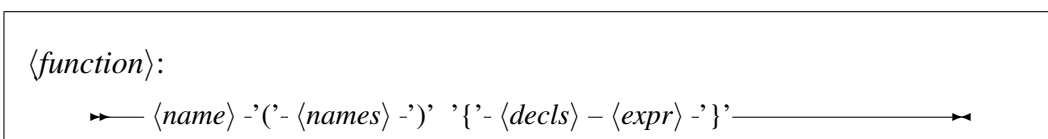
3.2.1 Listi yfir málrit

3.1	program	7
3.2	function	7
3.3	names	7
3.4	namelist	7
3.5	name	7
3.6	decls	8
3.7	expr	8
3.8	orExpr	8
3.9	andExpr	8
3.10	notExpr	8
3.11	opExpr	8
3.12	miniExpr	9
3.13	op	9
3.14	callFunc	9
3.15	argList	10
3.16	nonEmptyArgList	10
3.17	whileExpr	10
3.18	ifExpr	10
3.19	elseif	10
3.20	else	10
3.21	body	11
3.22	literal	11
3.23	int	11
3.24	digit	11
3.25	float	12
3.26	string	12
3.27	char	13
4.1	return	17

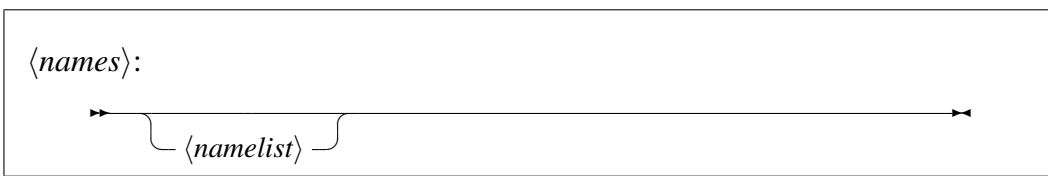
3.2.2 Málrít



Málrít 3.1: program



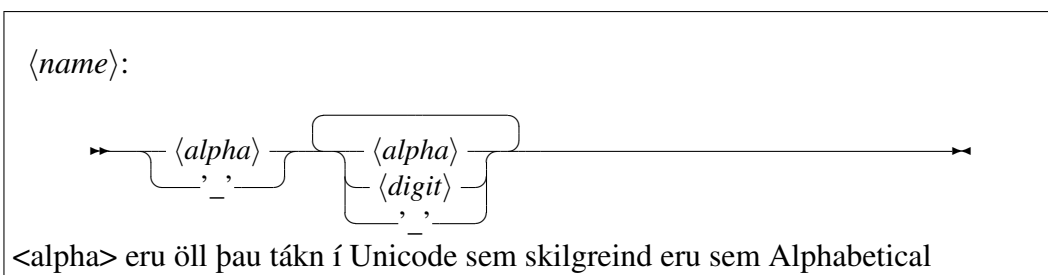
Málrít 3.2: function



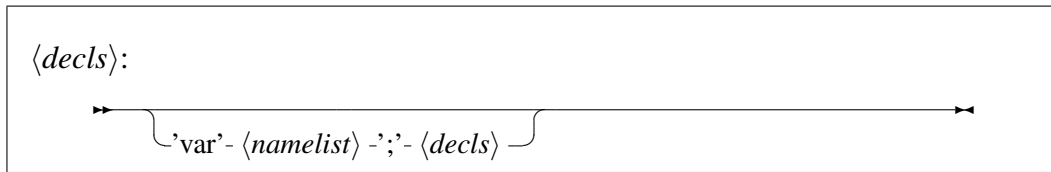
Málrít 3.3: names



Málrít 3.4: namelist



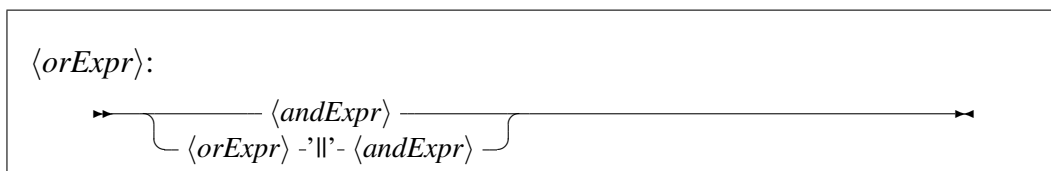
Málrít 3.5: name



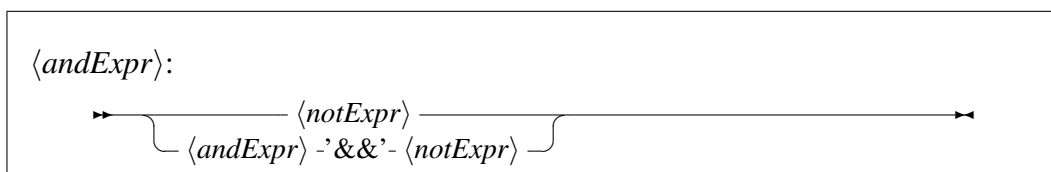
Málrit 3.6: decls



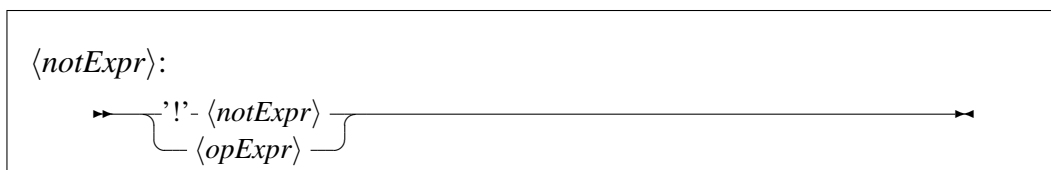
Málrit 3.7: expr



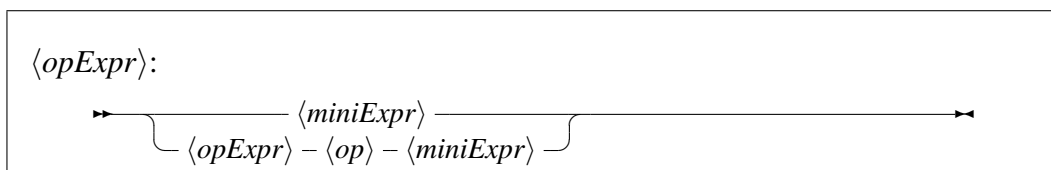
Málrit 3.8: orExpr



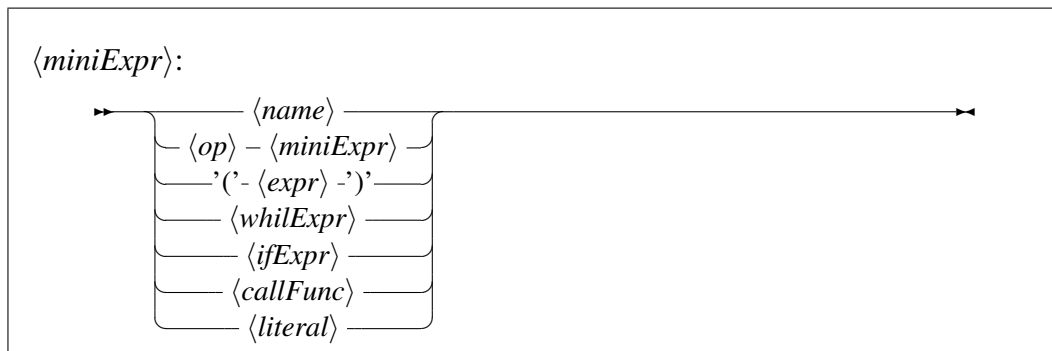
Málrit 3.9: andExpr



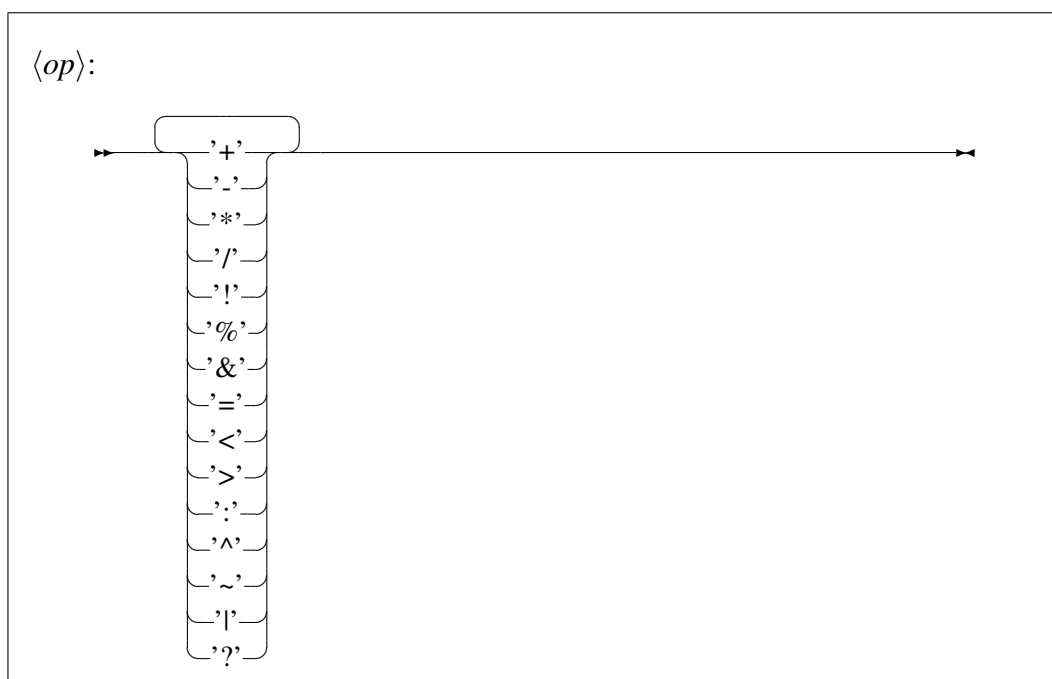
Málrit 3.10: notExpr



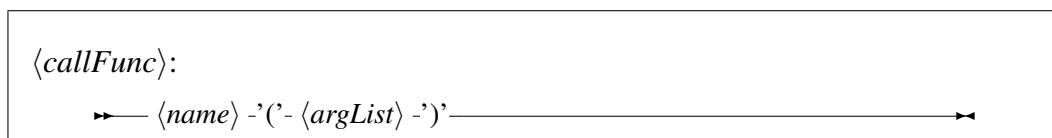
Málrit 3.11: opExpr



Málrit 3.12: miniExpr



Málrit 3.13: op



Málrit 3.14: callFunc

$\langle argList \rangle$:



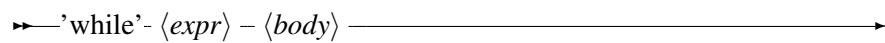
Málrit 3.15: argList

$\langle nonEmptyArgList \rangle$:



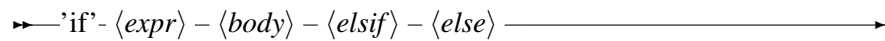
Málrit 3.16: nonEmptyArgList

$\langle whileExpr \rangle$:



Málrit 3.17: whileExpr

$\langle ifExpr \rangle$:



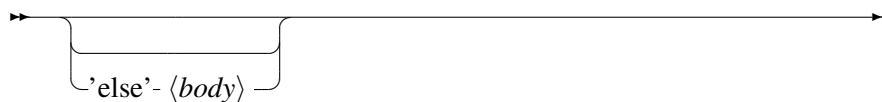
Málrit 3.18: ifExpr

$\langle elseif \rangle$:

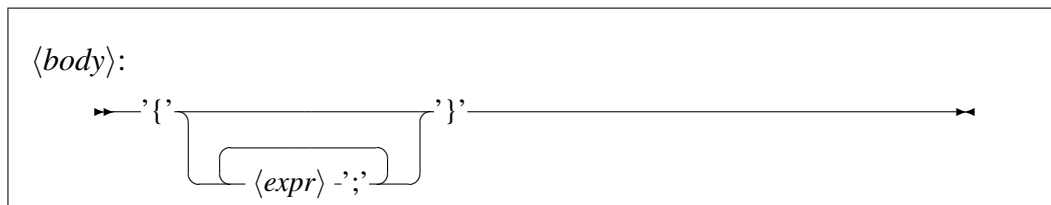


Málrit 3.19: elseif

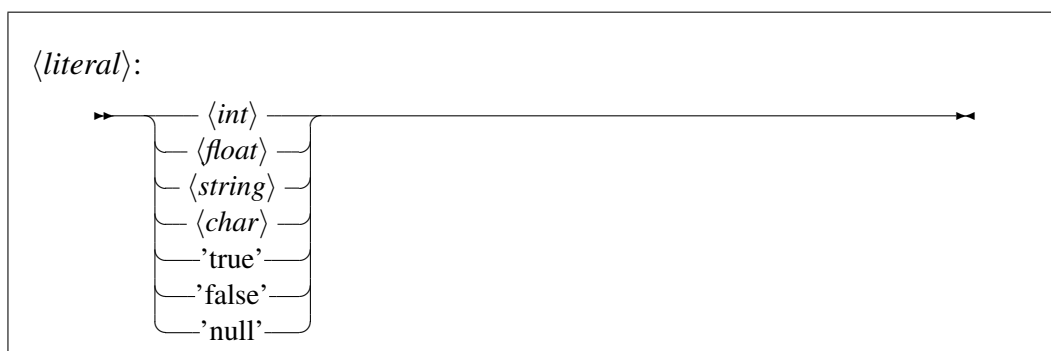
$\langle else \rangle$:



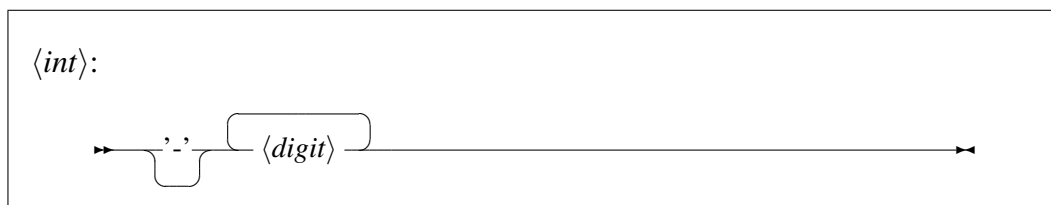
Málrit 3.20: else



Málrit 3.21: body



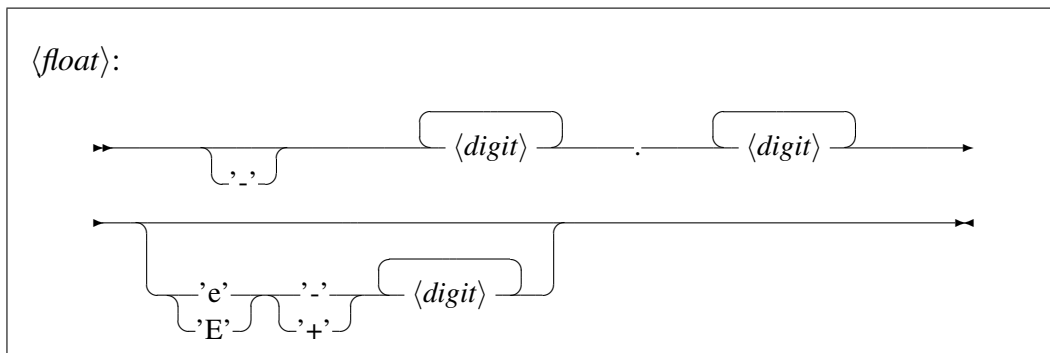
Málrit 3.22: literal



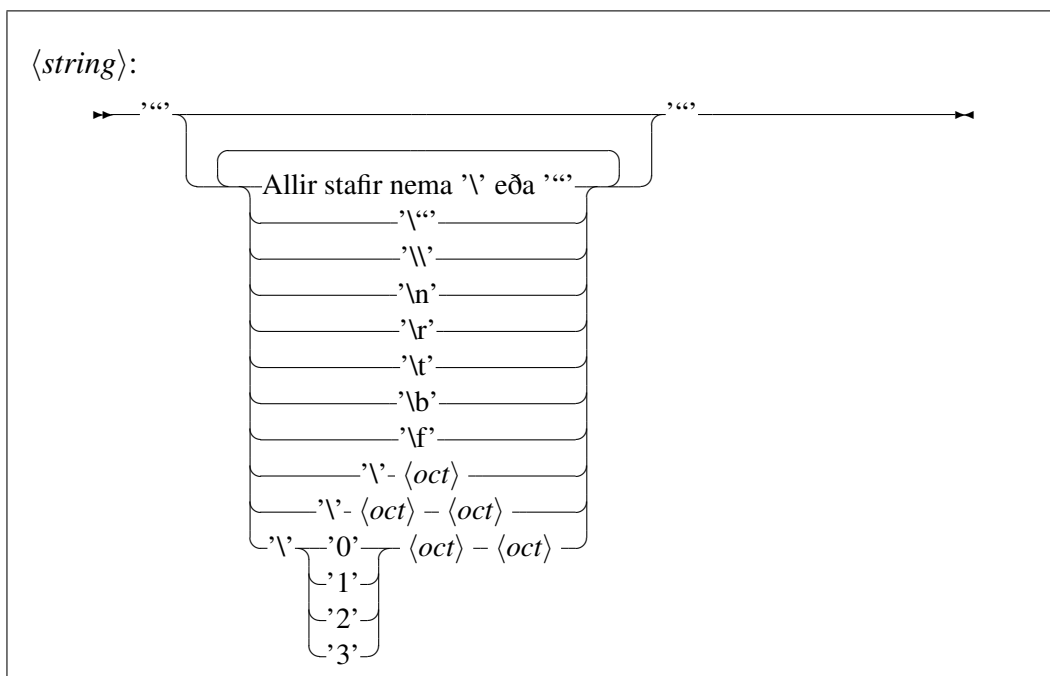
Málrit 3.23: int



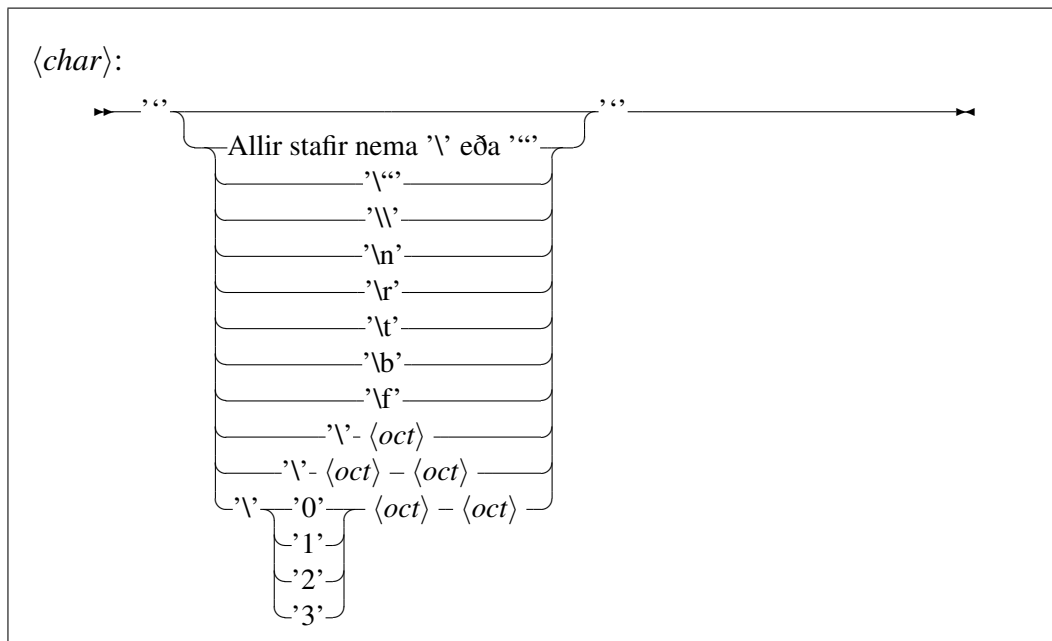
Málrit 3.24: digit



Málrit 3.25: float



Málrit 3.26: string



Málrit 3.27: char



4 Merking málsins

4.1 Mimo forrit

4.1.1 Einingar

Mimo forrit skiptist upp í eina eða fleiri einingar í sama skjali. Sjá málrit 3.1 á blaðsíðu 7 sem lýsir Mimo forriti. Hver eining í Mimo er skilgreining á falli. Sjá málrit 3.2 á blaðsíðu 7 sem sýnir uppbyggingu falla. Föll í Mimo geta skilað gildum og þannig haft samskipti sín á milli. Forrit 4.1 sýnir dæmi um fall sem tekur inn tvær heiltölur og skilar margfeldi þeirra.

```
margfeldi(x,y) {  
  var z;  
  z = x * y;  
  return z;  
}
```

Forrit 4.1: Dæmi um fall í Mimo

Eins og komið hefur fram þá þýðist Mimo forrit yfir á Morpho vélamál, þannig er þá hægt að nýta föll skilgreind í Mimo í Morpho forriti. Einnig þá býður Mimo sjálfkrafa uppá allar aðgerðir/föll sem BASIS einingin úr Morpho inniheldur, t.d samlagningu og margföldun, rökaðgerðir, úttaksaðgerðina writeln() og mikið meira.

4.1.2 Main() fallið

Til að forrit í Mimo sé keyranlegt eftir þýðingu þá þarf að skilgreina fallið main(). Main() fallið er skilgreint á sama hátt og önnur föll í Mimo. Þegar forritið er keyrt í Morpho þá leitar Morpho eftir main() fallinu og kallar á það í upphafi keyrslu.

```
main() {  
  writeln("Halló Heimur");  
}
```

Forrit 4.2: Halló Heimur forrit í Mimo

4.2 Gildi

Gildi í Mimo eru af sömu gerð og í Morpho, því forrit í Mimo er þýtt yfir á Morpho vélamál. Gildi í Mimo eru því tilvísanir í Java hluti. Mimo býður uppá heiltölur,

fleytitölur, strengi, stafi, rökgildin 'true' og 'false', ásamt tómri tilvisun 'null'. Sjá nánar um gildis-segðir í kafla 4.4 - Merking segða á blaðsíðu 15

4.3 Breytur

Í Mimo þarf ekki að segja til um týpur á breytum þegar þær eru skilgreindar þar sem að þær lúta svipuðum lögmálum og í Morpho. Breytur eru einungis staðværar en Mimo býður ekki uppá víðværar breytur. Skilgreina þarf breytur í upphafi falls áður en þær eru notaðar og gefið gildi (sjá málrit 3.2 á blaðsíðu 7). lykilorðið 'var' þarf að vera í upphafi hverrar línu sem breytur eru skilgreindar í. Forrit 4.3 gefur dæmi um hvernig og hvar í kóða breytur eru skilgreindar.

```
fall(x,y) {  
;;; Breytur skilgreindar  
var a,b,c;  
var h;  
var i;  
  
;;; Breytum gefið gildi  
a = 1;  
b = 2;  
c = a;  
}
```

Forrit 4.3: Skilgreining á breytum og gildisveiting

4.4 Merking segða

4.4.1 Heiltölusegð

Málrit 3.23 á blaðsíðu 11 lýsir málinu fyrir heiltölur (e. integer). Heiltölugildi eru samsvarandi heiltölugildum í Morpho og eru engin takmörk á stærð þeirra. Forrit 4.4 sýnir fáein dæmi um heiltölur.

```
123  
-321  
0  
1  
7876283  
-123432
```

Forrit 4.4: Dæmi um heiltölur

4.4.2 Fleytitölusegð

Málrit 3.25 á blaðsíðu 12 lýsir málinu fyrir fleytitölur (e. float). Fleytitölugildi eru samsvarandi fleytitölugildum í Morpho. Forrit 4.5 sýnir fáein dæmi um fleytitölur.

```
123.123
-321.34
0.345435
1.0
7876283.434
-123432.343
12.23e-12
```

Forrit 4.5: Dæmi um fleytitölur

4.4.3 Stafsegð

Málrit 3.27 á blaðsíðu 13 lýsir málinu fyrir stafi (e. char). Stafgildi eru samsvarandi stafgildum í Morpho. Forrit 4.6 sýnir fáein dæmi um stafi.

```
'a'
'b'
'e'
'R'
'4'
'\n'
```

Forrit 4.6: Dæmi um stafi

4.4.4 Strengsegð

Málrit 3.26 á blaðsíðu 12 lýsir málinu fyrir strengi (e. string). Gildi strengja eru samsvarandi gildi strengja í Morpho. Forrit 4.7 sýnir fáein dæmi um strengi.

```
"123"  
"-321"  
"Þetta er strengur"  
"Þetta\n er\n strengur sem spannar 3 línur"
```

Forrit 4.7: Dæmi um strengi

4.4.5 return-segð

Return segðin er notuð þegar hætta á í núverandi falli. Sé segðin *return x*; tilstaðar í falli þá skilar það gildi segðinnar *x*. Málrit 4.4.5 lýsir máli return-segðarinnar. Forrit 4.4.5 sýnir dæmi um notkun á return-segð í falli.

```
<return>:  
    ➡ 'return' - <expr> ————— ➡
```

Málrit 4.1: return

```
fall(x) {  
  var y;  
  y = 10;  
  return x + 10;  
}  
  
main() {  
  ;;; Skrifar 20 á aðalúttak  
  writeln(fall(10));  
}
```

Forrit 4.8: Dæmi um notkun á return

4.4.6 Kallsegð

Málrit 3.14 á blaðsíðu 9 ásamt málriti 3.15 og 3.16 lýsir málinu fyrir kallsegðir. Til að hægt sé að kalla á fall þarf það að vera skilgreint fyrst. Kafli 4.1.1 á blaðsíðu 14 fjallar m.a. um skilgreiningar á föllum. Föll geta tekið inn 0 eða fleiri segðir

sem viðföng. Ef við höfum kallsegðina $f(e0::eN)$ og $e0::eN$ eru segðir þá eru gildin reiknað úr $e0::eN$ og þau notuð sem viðföng í fallið f . Kallsegðin verður síðan að gildinu sem *return* segð f skilar, þ.e ef f inniheldur *return* segð. Forrit 4.4.6 sýnir einfalt dæmi um fallskall á fall sem skilar streng.

```
;;; Fallið 'f' skilgreint
f() {
  return "Halló Heimur";
}

main() {
  var x;
  ;;; Kallað á fallið 'f' í segð
  x = f();
  ;;; Nú er 'x' strengurinn "Halló Heimur"
}
```

Forrit 4.9: Dæmi um fallskall

4.4.7 Aðgerðir

Mimo býður upp á gildisveitinguna = ásamt almennum reikni- og samanburðar- aðgerðum sem finna má í öðrum málum.

Reikniaðgerðir

+ Samlagning

- Frádráttur

* Margföldun

/ Deiling

Samanburðaraðgerðir

< Minna en

> Stærri en

<= Minna en eða samasem

>= Stærri en eða samasem

== samasem

Einundaraðgerðir

- Mínus fyrir framan segð. Er jafngilt og að margfalda segðina með -1.

Forgangur Einundaraðgerðir hafa hæsta forgang, þar á eftir eru reiknaðgerðir og að lokum hafa samanburðaraðgerðir lægstan forgang. $*$ = $\%$ eru jafngild en hafa hærri forgang en $+$ —. Þær segðir sem eru innan sviga hafa hæðstan forgang.

4.4.8 Röksegðir

Lesfastarnir 'true' og 'false' gefa gildin satt og ósatt og samsvara rök gildum í Morpho. Einnig gefur lesfastinn 'null' gildið 'false'. Röksegðir gefa annaðhvort gildið 'true' eða 'false'. Mimo hefur þrjár röksegðir, $\&\&$ og $!$ sem standa fyrir 'og', 'eða' og 'ekki' rökaðgerðirnar Þeim er lýst ögn nákvæmar hér að neðan.

og-segð er röð segða með $\&\&$ á milli. Segðin skilar gildinu 'true' þá og því aðeins að allar hlutsegðir skili gildinu 'true'. Annars skilar og-segðin gildinu 'false'. Málrit 3.9 á blaðsíðu 8 lýsir málinu fyrir og-segðir. Séu p , q og r segðir þá eru segðirnar $p \&\& q \&\& r$, $(p \&\& q) \&\& r$ og $p \&\& (q \&\& r)$ jafngildar.

eða-segð er röð röksega með $||$ á milli. Segðin skilar gildinu 'false' þá og því aðeins að allar hlutsegðir skili gildinu 'false'. Annars skilar eða-segðin gildinu 'true'. Málrit 3.8 á blaðsíðu 8 lýsir málinu fyrir eða-segðir. Séu p , q og r segðir þá eru segðirnar $p || q || r$, $(p || q) || r$ og $p || (q || r)$ jafngildar.

ekki-segð er röksegð með $!$ fyrir framan. Segðin $!x$ skilar gildinu 'true' ef x skilar 'false' og 'false' skili x 'true'. Líkt og í Morpho þá eru 'null' og 'false' einu gildin sem eru ósönn. Þannig að $!null$ gefur 'true' en $!0$ gefur 'false' Málrit 3.10 á blaðsíðu 8 lýsir málinu fyrir ekki-segð.

4.4.9 if-segð

Málrit 3.18 á blaðsíðu 10 lýsir málinu fyrir if-segðir. If-segð er notuð til að meta segð og bregðast við eftir því hvort segðin skili 'true' eða 'false'.

```
if (e) {  
    ;;; B1  
} else {  
    ;;; B2  
};
```

Forrit 4.10: Einföld if-segð

Forrit 4.4.9 sýnir kóða fyrir if-segðina $if(e)\{B1\}else\{B2\}$. Ef e er 'true' þá er $B1$ framkvæmt, annars er $B2$ framkvæmt. Ekki þarf að enda allar if setningar á $else$ segð. Einnig er valfrjálst að nota $elseif$ sem er jafngilt því og að hreiðra if segð inni $else$.

4.4.10 while-segð

Málrit 3.17 á blaðsíðu 10 lýsir málinu fyrir while-segð. While-segð er eina segðin í Mimo sem býður uppá endurtekningu aðgerða. Virkni segðarinnar $while(p)B1$ má lýsa sem svo að $B1$ er endurtekið þangað til p er 'false'. Forrit 4.4.10 sýnir einfalt dæmi um while segð í kóða.

```
while(expression) {  
    ;;; B1  
};
```

Forrit 4.11: Dæmi um while-segð