

Handbók fyrir NanoMorpho

Pétur Daníel Ámundason

26. apríl 2018

Efnisyfirlit

1	Inngangur	3
2	Notkun og uppsetning	3
3	Málfræði	4
3.1	Frumeiningar málsins	4
3.1.1	Athugasemd	4
3.1.2	Lykilorð	4
3.2	Mállýsing	4
3.2.1	Forrit	4
3.2.2	Föll	4
3.2.3	Stofnar	4
3.2.4	Segðir	5
4	Merking málsins	5
4.1	Breytur	6
4.2	Merking segða	6
4.2.1	null-segð	6
4.2.2	true-false segð	6
4.2.3	false-segð	6
4.2.4	Heiltölusegð	6
4.2.5	Fleytitölusegð	6
4.2.6	Stafsegð	6
4.2.7	Strengsegð	6
4.2.8	return-segð	7
4.2.9	Röksegðir	7
4.2.10	Kallsegð	7
4.2.11	Tvíundaraðgerðir	7
4.2.12	Einundaraðgerðir	7
4.2.13	if-segð	8
4.2.14	while-segð	8
4.3	Föll og forrit	8

1 Inngangur

Enn einn þýðandi NanoMorpho Einföld útgáfa af Morpho. NanoMorpho er bálkmótað mál.

2 Notkun og uppsetning

Uppsetning Notast skal við Git til þess að sækja þýðandan.

Git \$ Git clone https://Github.com/rutep/NanoMorpho.Git

Þegar náð hefur verið í skrár þá eru þær þrjár skrár. **NanoMorphoLexer.java** sem er lesgreinir, **NanoMorphoParser.java** og **NanoMorphoParserVal.java** sem sjá um þáttun og milli þulusmíði.

Morpho Morphoskrárnar eru til þess að geta þýtt **NanoMorpho** skrár yfir á Morpho smalarmál og smalarmálið yfir í keyrslu hæfa Morphoskrá. Þegar fyrrnefnda er búið þá er hægt að keyra forritið í gegnum Morpho.

makefile Hér fyrir neðan sést makefile skrá sem hægt er að nota til þess að auðvelda sér þýðingu og keyrslu **NanoMorpho** forrits. Ef búnaður er ekki til staðar til þess að nota makefile skrá þá er hægt að keyra beint skipanir á skipunarlínu ef java þýðandi og Morpho er upp sett.

- \$ javac NanoMorphoLexer.java NanoMorphoParser.java NanoMorphoParserVal.java
- Skipuninn að ofan þýðir java skrár
- \$ java NanoMorphoParser test.s > test.masm
- Býr til Morpho smalarmáls skrá test.masm
- \$ Morpho -c test.masm
- \$ Morpho test
- Býr til test.mexem skrá og svo keyrir forrit

`SHELL=/usr/bin/env /bin/bash`

```
parser: NanoMorphoParser.java NanoMorphoParserVal.java NanoMorphoLexer.java
javac NanoMorphoLexer.java NanoMorphoParser.java NanoMorphoParserVal.java
```

```
test: NanoMorphoParser.class NanoMorphoParserVal.class
java NanoMorphoParser test.s > test.masm
```

```
compile: test.masm
Morpho -c test.masm
```

```
run: test.mexe
Morpho test
```

3 Málfræði

3.1 Frumeiningar málsins

Samhengislaust mál Línubil og línu endingar hafa enga merkingu fyrir þýðandan og væri hægt að skrifa heilu forritin í einni línu með engum línubilum.

3.1.1 Athugasemd

Dæmi um athugasemd `;;; Hunsað af þýðanda Þýðandinn mun hunsa allt sem kemur á eftir athugasemdar í þeirri línu sem hún er.`

3.1.2 Lykilorð

Lykilorðin í þýðandanum eru:

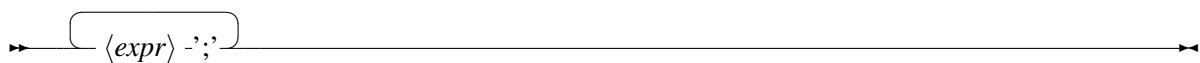
`else, elsif, false, if, null, return, true, var, while, println`

3.2 Mállýsing

$\langle idlist \rangle$:



$\langle exprs \rangle$:



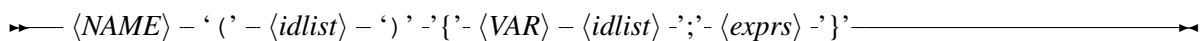
3.2.1 Forrit

$\langle program \rangle$:



3.2.2 Föll

$\langle function \rangle$:



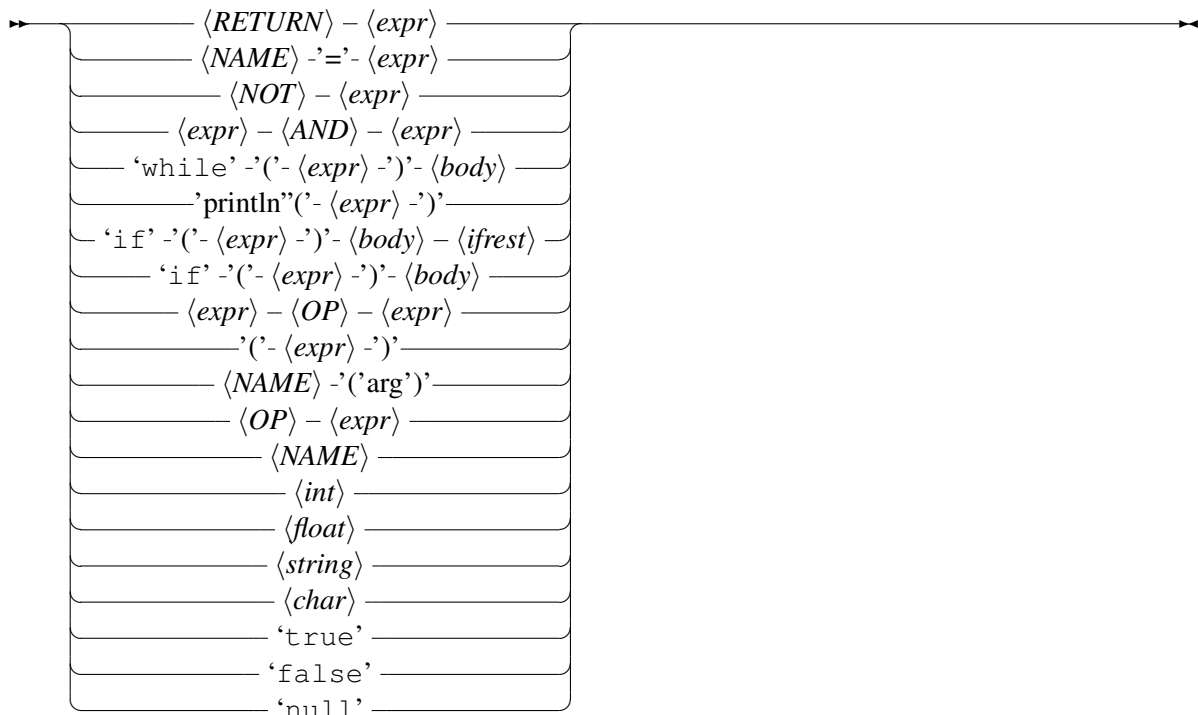
3.2.3 Stofnar

$\langle body \rangle$:



3.2.4 Segðir

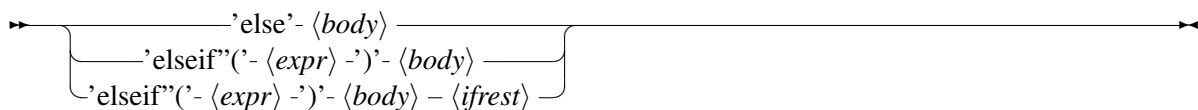
$\langle expr \rangle$:



$\langle arg \rangle$:



$\langle ifrest \rangle$:



4 Merking málsins

NanoMorpho skiptis upp í að það er eitt main fall sem keyrt er og úr því er hægt að kalla á önnur skilgreind föll.

```

foo () {
    var a;
    a = 1;
    return a;
}

main () {
    var a;
    a = foo ();
}

```

4.1 Breytur

Fyrsta gildisveitinginn sem á sér stað er þegar breyta er búinn til, þá mun hún innihalda null gildið. Seinna meir í stofni falls er hægt að gefa henni nýtt gildi. Notkunardæmi er hér fyrir neðan. Breytunöfn verða hafa a.m.k. einn bókstaf í sér og geta innihaldið tölur.

```
main() {  
    var a;    ;;; a = null  
    a = 0;    ;;; a = 0  
}
```

4.2 Merking segða

4.2.1 null-segð

Þjónar þeim tilgangi að sýna fram á að breyta hafi ekki verið gefið neitt gildi.

4.2.2 true-false segð

Boolean gildi true notað í boolean reikningi.

4.2.3 false-segð

Boolean gildi false notað í boolean reikningi.

4.2.4 Heiltölusegð

Heiltölu gildi.

4.2.5 Fleytitölusegð

Fleytitölu gildi.

4.2.6 Stafsegð

Stafsegðar gildi 's'.

4.2.7 Strengsegð

Strengsegðar gildi "Strengsegð".

4.2.8 return-segð

Return skilar gildi úr falli. Notkunardæmi:

```
foo() {  
    return 1;           ;;; Skilar 1 ef kallað er á  
}
```

4.2.9 Röksegðir

Röksegðir eru or ||, and &&, !=, >=, <= og == .

Þær eru að mestu leiti reiknaðar frá vinstri til hægri. Notkunardæmi:

```
1 != 2           ;;; gefur  
true || false   ;;; true  
true && true     ;;; true  
1 >= 1          ;;; true  
1 <= 1          ;;; true  
1 == 1          ;;; true
```

4.2.10 Kallsegð

Fall eru annað hvort úr basis eða gert af notenda. Fall tekur inn breytur sem viðföng og reikningur úr þeim er frá vinstri til hægri. Dæmi um slíkt fall sem notendi getur gert og notkun:

```
foo(a, b, s) {  
    var t;  
    return 1;  
}  
  
main() {  
    var x;  
    x = foo(1+1, 2-5, -3);    ;;; foo fall skilar einum  
}
```

4.2.11 Tvíundaraðgerðir

Tvíundaraðgerðir eru á infix formi.

4.2.12 Einundaraðgerðir

Einundaraðgerðir hafa hæsta forgang.

4.2.13 if-segð

Hér er dæmi um notkun á if elsif else segðum. Ef s1 er satt þá body1. Ef s1 ósatt og s2 satt þá body2. Ef s1 og s2 ósatt þá body3.

```
main() {
    var s1, s2, s3;
    if(s1){
        ;;; body1
    } elsif (s2) {
        ;;; body2
    } else {
        ;;; body3
    };
}
```

4.2.14 while-segð

while-segðin mun alltaf keyrast ef ástandið s1 helst satt. Dæmi um while-segð. Í gefni while-segð ef röksegðini s1 helst sönn þá mun fyrir hvert stef í while lykkju i hækka um einn.

```
main() {
    var i;
    i = 0;
    while(s1){
        i = i + 1;
    };
}
```

4.3 Föll og forrit

Hér er dæmi um heilt NanoMorpho forrit. Þetta forrit reiknar fyrstu 12 Fibonacci tölurnar.

```
;;; Fibonacci
fibonacci(n){
    var ;
    if(n < 0) {
        return n * -1;
    } elsif( n == 0) {
        return n * ( 1 - 2);
    } else {
        return fibonacci(n-1) + fibonacci(n-2);
    };
}
main(){
    var n;
    n = 0;
    while(n < 12){
        n = n+1;
        println(fibonacci(n)); }; }
```