

Handbók fyrir Kaladin

Matthías Páll Gissurarson

21. apríl 2014

Útdráttur

Kaladin er mál sem reynir að vera eins skýrt og það getur verið, en er heldur einfalt hvað forritunarmál varða. Kaladin keyrir á Morpho sýndarvélinni. Kaldin er “eager”, “procedural” mál, en það er bálkmótað hvað föll varðar. Kaladin styður ekki “global” breytur.

Efnisyfirlit

1	Inngangur	2
2	Notkun og uppsetning	4
3	Málfræði	4
3.1	Frumeiningar málsins	4
3.1.1	Athugasemdir	4
3.1.2	Lykilorð	4
3.2	Mállýsing	5
4	Merking málsins	7
4.1	Gildi	7
4.2	Breytur	7
4.3	Merking segða	7

4.3.1	Heiltölusegð	7
4.3.2	Fleytitölusegð	7
4.3.3	Stafsegð	7
4.3.4	Strengsegð	7
4.3.5	Listasegð	7
4.3.6	return-segð	7
4.3.7	Röksegðir	7
4.3.8	Kallsegð	7
4.3.9	Tvíundaraðgerðir	7
4.3.10	Einundaraðgerðir	7
4.3.11	if-segð	7
4.3.12	while-segð	7
5	Mállýsing	7
5.1	Ýmis málrit með hjálp syntax pakkans í LaTeX	7
5.1.1	Forrit	8
5.1.2	Föll	8
5.1.3	Stofnar	8
5.1.4	Skilgreiningar	9
5.1.5	Segðir	9
5.1.6	Annar möguleiki	10

1 Inngangur

Kaladin er mál sem reynir að vera eins skýrt og það getur verið (verbose) og keyrir á Morpho sýndarvélinni.

Kaladin má nota frá netinu í gegnum síðuna `kaladin.mpg.is`, en til þess að geta keyrt skrár sem málið skilar og þýða án nettengingar, þá þarf að sækja

málið frá github. Málið og handbók þessa má finna á <https://github.com/Tritlo/Kaladin>.

Kaladin þýðist yfir í Morpho smalamál. Til þess að geta notað Kaladin, þá þarf Morpho að vera uppsett á tölvunni eða `morpho.jar` skráin að vera til staðar. Morpho má sækja á <http://morpho.cs.hi.is/>, en höfundur Morpho er Snorri Agnarsson.

Kaladin þýðandinn er skrifaður í JavaScript, en til þess að þýða Kaladin skrár (oftast með `.kal` endingu), þá þarf `Node.js` að vera uppsett á tölvunni þar sem þýða á skránnar, en `Node.js` gerir manni kleyft að keyra JavaScript án vafra. Upplýsingar um `Node.js` og uppsetningu þess má finna á <http://nodejs.org/>.

Kaladin er að mörgu leyti líkt Morpho, og getur kallað á öll föll sem Morpho býður uppá, án þess að krefjast sérstakrar tiltekningar á því.

Einfalt dæmi um fall er:

```
1 #Pre : x is int >= 0
2 #Use : f = fibo(x);
3 #Post: f is the x-th fibonacci number
4 The fibo function, whose argument is x, is as follows:
5     if x is 0 then
6         return 0
7     else if x is 1 then
8         return 1
9     else
10         return fibo(x-1) + fibo(x-2)...
11 which is what we wanted to do.
```

en þetta fall má einnig skrifa á hátt sem nýtir sér að segðir hafa gildi, þ.e.

```
1 #Pre : x is int >= 0
2 #Use : f = fib(x);
3 #Post: f is the x-th fibonacci number
4 The fib function, whose argument is x, is as follows:
5     return if x is less than 2 then x else return fib(
6         x-1) + fib(x-2)..
6 which is what we wanted to do.
```

2 Notkun og uppsetning

Til þess að þýða Kaladin skrá er notuð eftirfarandi skipun úr möppunni þar sem `kaladin.js` er í (hér er “initial” notað sem nafnið á skránni sem þýða á).

```
node kaladin.js initial.kal
```

en skipunin býr til skjal sem ber heitið “initial.masm” í sömu möppu og skipunin var keyrð úr, sem inniheldur smalamál fyrir Morpho sýndarvélina.

Skránni má svo þýða með Morpho með skipuninni

```
java -jar morpho.jar -c initial.masm
```

og svo keyra með

```
java -jar morpho.jar initial
```

3 Málfræði

3.1 Frumeiningar málsins

3.1.1 Athugasemdir

3.1.2 Lykilorð

Lykilorð í Kaladin eru frekar mörg, en sum lykilorðin eru tvö eða fleiri orð hlið við hlið. Lykilorðin eru eftirfarandi: “function”, “While”, “while”, “If”, “if”, “else”, “and”, “or”, “not”, “, whose arguments are”, “, whose argument is”, “, which takes no arguments”, “, is as follows:”, “true”, “and return ”, “True”, “false”, “False”, “Return”, “return”, “We let” “we let”, “Let”, “let”, “then”, “The”, “the”, “Do”, “do”, “None”, “in”, “be”, “which is what we wanted to do.”, “is less than or equal to”, “is greater than or equal to”, “is less than”, “is greater than”, “is”, “, and”.

Einnig eru táknin “.”, “, ”, “; ”, “: ”, “==”, “=”, “^”, “*”, “/”, “++”, “-”, “+”, “(”, “)” frátekin.

3.2 Mállýsing

$\langle program \rangle ::= \langle functions \rangle$

$\langle functions \rangle ::= \langle functions \rangle \langle function \rangle \text{EXPRDELIM}$
 $\quad | \quad \langle function \rangle \text{EXPRDELIM}$

$\langle posArgDecl \rangle ::= \text{NOARGSDECL}$
 $\quad | \quad \text{ARGSDECL } \langle names \rangle$
 $\quad | \quad \text{ONEARGDECL NAME}$

$\langle function \rangle ::= \text{THE NAME DEF } \langle posArgDecl \rangle \text{FUNDECLEND } \langle optdecls \rangle$
 $\quad \langle optexprs \rangle$

$\langle names \rangle ::= \langle names \rangle ', ' \text{NAME}$
 $\quad | \quad \text{NAME}$

$\langle optnames \rangle ::= \lambda$
 $\quad | \quad \langle names \rangle$

$\langle args \rangle ::= \langle args \rangle ', ' \langle expr \rangle$
 $\quad | \quad \langle expr \rangle$

$\langle optargs \rangle ::= \lambda$
 $\quad | \quad \langle args \rangle;$

$\langle body \rangle ::= \langle exprlist \rangle$

$\langle decl \rangle ::= \text{VAR NAME} = \langle expr \rangle$

$\langle optdecls \rangle ::= \lambda$
 $\quad | \quad \langle decls \rangle$

$\langle decls \rangle ::= \langle decls \rangle \langle decl \rangle ', '$
 $\quad | \quad \langle decl \rangle ', '$

$\langle exprlist \rangle ::= \text{exprlist } ', ' \langle expr \rangle | \langle expr \rangle$

$\langle ifrest \rangle ::= \text{ELSE } \langle body \rangle$

$\langle ifst \rangle ::= \text{IF expr THEN } \langle body \rangle \langle ifrest \rangle \text{EXPRDELIM}$

$\langle whilest \rangle ::= \text{WHILE } \langle expr \rangle \text{ THEN } \langle body \rangle \text{EXPRDELIM}$

$$\begin{aligned}
\langle expr \rangle ::= & \langle expr \rangle '+' \langle expr \rangle \\
& | \langle expr \rangle '-' \langle expr \rangle \\
& | \langle expr \rangle '*' \langle expr \rangle \\
& | \langle expr \rangle '/' \langle expr \rangle \\
& | \langle expr \rangle '^' \langle expr \rangle \\
& | '-' \langle expr \rangle \\
& | \langle expr \rangle '<=' \langle expr \rangle \\
& | \langle expr \rangle '>=' \langle expr \rangle \\
& | \langle expr \rangle '<' \langle expr \rangle \\
& | \langle expr \rangle '>' \langle expr \rangle \\
& | \langle expr \rangle '==' \langle expr \rangle \\
& | \langle expr \rangle '++' \langle expr \rangle \\
& | NAME '(' optargs ')' \\
& | \langle expr \rangle \text{ AND } \langle expr \rangle \\
& | \langle expr \rangle \text{ OR } \langle expr \rangle \\
& | \text{ NOT } \langle expr \rangle \\
& | \text{ LET NAME '=' } \langle expr \rangle \\
& | NAME '=' \langle expr \rangle \\
& | NAME \\
& | \text{ RETURN } \langle expr \rangle \\
& | \text{ NONE} \\
& | \text{ STRING} \\
& | \text{ NUMBER} \\
& | \text{ TRUE} \\
& | \text{ FALSE} \\
& | '(' \langle expr \rangle ')'' \\
& | \text{ ifst} \\
& | \text{ whilest} \\
\\
\langle exprs \rangle ::= & \langle exprs \rangle \langle expr \rangle \text{ EXPRDELIM} \\
& | \langle expr \rangle \text{ EXPRDELIM} \\
\\
\langle optexprs \rangle ::= & \lambda \\
& | \langle exprs \rangle
\end{aligned}$$

4 Merking málsins

4.1 Gildi

4.2 Breytur

4.3 Merking segða

4.3.1 Heiltölusegð

4.3.2 Fleytitölusegð

4.3.3 Stafsegð

4.3.4 Strengsegð

4.3.5 Listasegð

4.3.6 return-segð

4.3.7 Röksegðir

4.3.8 Kallsegð

4.3.9 Tvíundaraðgerðir

4.3.10 Einundaraðgerðir

4.3.11 if-segð

4.3.12 while-segð

5 Mállýsing

5.1 Ýmis málrit með hjálp `syntax` pakkans í LaTeX

Hér eru nokkur málrit skilgreind með hjálp `syntax` pakkans. Breytið þessu eftir þörfum, eða notið aðrar aðferðir ef það hentar ykkur betur.

Finna má upplýsingar um `syntax` pakkann á vefnum¹.

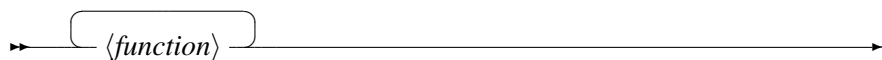
Takið eftir að ráðlegt er að gera einhverjar breytingar hér, þó ekki væri til annars en að gera textalýsingar á merkingu einfaldari með tilliti til forgangs ýmissa mál fyrirbæra.

$\langle id_list \rangle$:



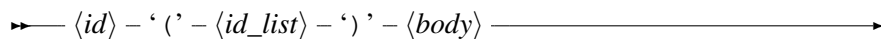
5.1.1 Forrit

$\langle program \rangle$:



5.1.2 Föll

$\langle function \rangle$:



5.1.3 Stofnar

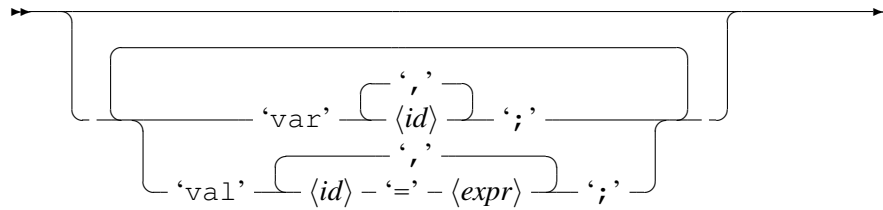
$\langle body \rangle$:



¹<http://www.tug.org/texlive/Contents/live/texmf-dist/doc/latex/mdwtools/syntax.pdf>

5.1.4 Skilgreiningar

$\langle decls \rangle$:

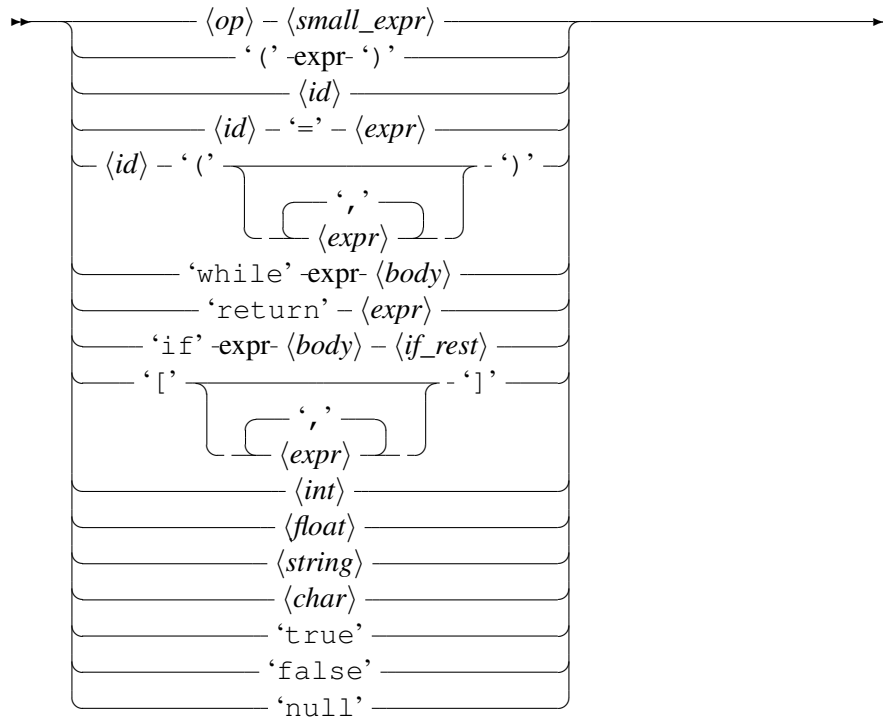


5.1.5 Segðir

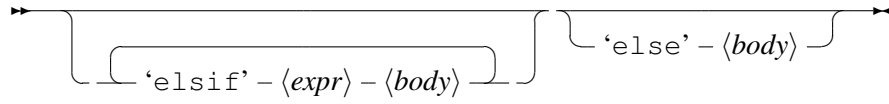
$\langle expr \rangle$:



$\langle small_expr \rangle$:

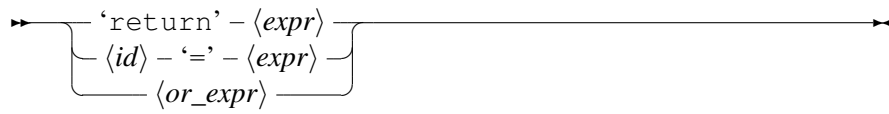


$\langle if_rest \rangle$:



5.1.6 Annar möguleiki

$\langle expr \rangle$:



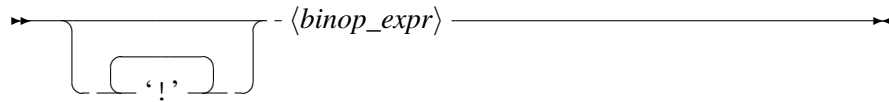
$\langle or_expr \rangle$:



$\langle and_expr \rangle$:



$\langle not_expr \rangle$:



$\langle binop_expr \rangle$:



$\langle small_expr \rangle:$ 