

Handbók fyrir Kaladin

Matthías Páll Gissurarson

22. 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 ólatt skipanamál, en það er bálkmótað hvað föll varðar. Kaladin styður staðværar breytur.

Efnisyfirlit

1	Inngangur	2
2	Notkun og uppsetning	3
3	Málfræði	4
3.1	Frumeiningar málsins	4
3.1.1	Athugasemdir	4
3.1.2	Lykilorð	4
3.2	Mállýsing	4
3.2.1	Tákn	4
3.2.2	BNF mállýsing	6

4	Merking málsins	8
4.1	Gildi	8
4.2	Breytur	9
4.3	Merking segða	9
4.3.1	Bókstafssegðir	9
4.3.2	return-segð	9
4.3.3	Röksegðir	10
4.3.4	Kallsegð	10
4.3.5	Tvíundaraðgerðir	10
4.3.6	Einundaraðgerðir	10
4.3.7	if-segð	11
4.3.8	while-segð	11

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 skrárnar, 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.

Skránna 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

Athugum að fyrst að mörg lykilorð má skrifa með bæði stórum og litlum upphafsstaf, og er merkingin sú sama hvora leiðina sem farin er. Bil skipta ekki mál, nema hvað að bil í lykilorðum verða að vera eins og bilin eru í lykilorðum í listanum sem kemur hér á eftir.

Ef táknið # kemur fyrir í byrjun línu, þá er allt sem er í línunni á eftir því tákni hent og ekki tekið til greina.

3.1.2 Lykilorð

Lykilorð í Kaladin eru frekar mörg, en sum lykilorðin eru tvö eða fleiri orð hlið við hlið. Ef svo er, þá verða þau að vera skrifuð nákvæmlega eins, þ.e. ekki má bæta við auka bili milli orðanna. 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”, “true”, “false”, “False”, “Return”, “return”, “We let” “we let”, “Let”, “let”, “then”, “The”, “the”, “Then we do”, “then we do”, “We do”, “we do”, “then we”, “Then we”, “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

3.2.1 Tákn

Í mállýsingu eru tákn innan oddklofa millitákn, tákn sem eru innan ' eru lokatákn, og tákn sem eru með stórum stöfum eru einnig lokatákn, en merking þeirra er gefin í eftirfarandi töflu:

Regluleg segð sem lýsir lokatákni í forritskóða	Tákn í mállýsingu
"function"	DEF
"While"	WHILE
"while"	WHILE
"If"	IF
"if"	IF
"else"	ELSE
"and"	AND
"or"	OR
"not"	NOT
", whose arguments are"	ARGSDECL
", whose argument is"	ONEARGDECL
", which takes no arguments"	NOARGSDECL
", is as follows:"	FUNCDECLEND
"true"	TRUE
"True"	TRUE
"false"	FALSE
"False"	FALSE
"Return"	RETURN
"return"	RETURN
"We let"	VAR
"we let"	VAR
"Let"	LET
"let"	LET
"then we do"	TDO
"Then we do"	TDO
"we do"	TDO
"We do"	TDO
"Then we"	TDO
"then we"	TDO

Regluleg segð sem lýsir lokatákni í forritskóða	Tákn í mállýsingu
"then"	THEN
"The"	THE
"the"	THE
"Do"	DO
"do"	DO
"None"	NONE
"in"	IN
"be"	=
"which is what we wanted to do."	EXPRDELIM
"is less than or equal to"	<=
"is greater than or equal to"	>=
"is less than"	<
"is greater than"	>
"is"	==
", and"	,
" ("	(
") ")
"="	=
"*"	*
"/"	/
"++"	++
"_"	-
"+"	+
" , "	,
\ "[^\\"]*\\"	STRING
\ '[^\\']*\\'	STRING
[A-Za-z]([A-Za-z0-9])*	NAME
[0-9]+(("[0-9"])?\b	NUMBER
\:	THEN
\;	EXPRDELIM
\.	EXPRDELIM

3.2.2 BNF mállýsing

$\langle \text{program} \rangle ::= \langle \text{functions} \rangle$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\langle expr \rangle ::= \langle expr \rangle '+ ' \langle expr \rangle$
 $\quad | \langle expr \rangle '- ' \langle expr \rangle$
 $\quad | \langle expr \rangle '* ' \langle expr \rangle$
 $\quad | \langle expr \rangle '/ ' \langle expr \rangle$
 $\quad | \langle expr \rangle '^ ' \langle expr \rangle$
 $\quad | '- ' \langle expr \rangle$

```

| <expr> '<=' <expr>
| <expr> '>=' <expr>
| <expr> '<' <expr>
| <expr> '>' <expr>
| <expr> '==' <expr>
| <expr> ++ <expr>
| NAME '(' optargs ')'
| <expr> AND <expr>
| <expr> OR <expr>
| NOT <expr>
| LET NAME '=' <expr>
| NAME '=' <expr>
| NAME
| RETURN <expr>
| NONE
| STRING
| NUMBER
| TRUE
| FALSE
| '(' <expr> ')'
| ifst
| whilest

<exprs> ::= <exprs> <expr> EXPRDELIM
| <expr> EXPRDELIM

<optexprs> ::= λ
| <exprs>

```

4 Merking málsins

4.1 Gildi

Gildi í Kaladin geta verið strengir, heiltölur, fleytitölur, True, False og None.

4.2 Breytur

Breytur í kaladin er skilgreindar með því að segja `We let x be expr`, þar sem að `expr` er einhver lögleg Kaladin segð.

Hægt er að breyta gildi breytu með því að segja `x = expr` eða `Let x be expr` eða `Let x = expr`.

4.3 Merking segða

4.3.1 Bókstafssegðir

None-segð `None` segð hefur sömu merkingu og `null` í Morpho.

Heiltölusegð Heiltölu segð er segð á forminu `a`, þar sem `a` má vera hvaða heiltala sem er, og gildi hennar er heiltalan `a`. Hún fær sama gildi og samsvarandi segð í Morpho.

Fleytitölusegð Fleytitölu segð er segð á forminu `b`, þar sem `b` má vera hvaða fleytitala sem er, og gildi hennar er fleytitalan `b`. Hún fær sama gildi og samsvarandi segð í Morpho.

Strengsegð Strengsegð er runa stafa, núll eða fleiri, sem afmarkaðir eru með `"` eða `'`. Hún fær sama gildi og samsvarandi segð í Morpho.

Sanngildissegð `True` hefur sömu merkingu og `true` í Morpho. `False` hefur sömu merkingu og `false` í Morpho.

4.3.2 return-segð

Return segðir eru á forminu `Return expr`, þar sem að `expr` er einhver lögleg Kaladin segð.

Return segðir geta aðeins komið fyrir inni í fallsskilgreiningum, en þegar að Return segð kemur fyrir í keyrslu, þá er reiknað út hvert gildi `expr` er, og það gildi sett sem skilagildi fallsins, og fallið látið skila.

4.3.3 Röksegðir

Eftirfarandi samanburðar segðir eru í Kaladin, “is less than or equal to”, “is greater than or equal to”, “is less than”, “is greater than” og “is”, en þær haga sér eins og \leq , \geq , $<$, $>$ og $==$ haga sér í Morpho (og er í raun bara kall á samsvarandi Morpho fall).

Einnig eru eftirfarandi rökvirkjar, “and”, “or” og “not”. Þeir haga sér eins og rökvirkjarnir \wedge , \vee og \neg , en þeir skammhleypa, þ.e. að þegar þeir koma að gildi sem mundi láta gildi segðarinnar vera ótvírætt ákvarðaða, þá er hætt að meta gildi segðanna, og það gildi sem ákvarðað var verður gildi segðarinnar.

Sanngildi segða í Kaladin er eins og sanngildi segða í Morpho, nema í Kaladin hefur None sama gildi og null.

4.3.4 Kallsegð

Kall segð í Kaladin er á forminu “posVerbose f(x₁, . . . , x_n)”.

Þar sem posVerbose er “We do”, eða “Then we do” eða “Then we” eða ekkert.

Þar sem f er fall skilgreint í sömu skrá, eða fall úr Morpho. Athugið að fjöldi inntaksstærða í fallið verður að sú sama og fallið er skilgreint fyrir.

4.3.5 Tvíundaraðgerðir

Eftirfarandi tvíundaraðgerðir eru í Kaladin: “*”, “/”, “-”, “+”, “++”, auk þeirra sem koma fyrir í röksegðum.

Tvíundaraðgerðirnar virka eins og þær gera í Morpho og eru í raun

4.3.6 Einundaraðgerðir

Í Kaladin eru tvær einundaraðgerðir, en það er not virkinn, sem breytir sanngildi segðar yfir í öfugt sanngildi. – einundarvirkinn er kall á samsvarandi virkja í Morpho.

4.3.7 if-segð

if-segð er á forminu `If expr then body1 else body2.` eða `If expr then do body1 else body2.`

þar sem `body1` og `body2` er listi af segðum aðskilin með kommu þ.e. `e1, e2, e3,` and `e4` eða bara `e1, e2, e3,` `e4`.

Ef sanngildi `expr` er satt, þá er `body1` framkvæmdur, þ.e. segðirnar í `body1` eru gildaðar í þeirri röð sem þær koma fyrir. Annars er `body2` framkvæmdir.

4.3.8 while-segð

while-segð er á forminu `while expr then body.` eða `while expr then do body.`

þar sem `body` er listi af segðum aðskilin með kommu þ.e. `e1, e2, e3,` and `e4` eða bara `e1, e2, e3,` `e4`.

Á meðan sanngildi `expr` er satt, þá er `body` framkvæmt. Athugið að sanngildi `while` segðar er alltaf `false`.