TÖL304G
Forritunarmál
Verkefnablað 3
Assignment Sheet 3

Snorri Agnarsson

6. september 2022

Hópverkefni — Group Assignments

Íhugið eftirfarandi λ-segðir. Consider the following λ-expressions.

- Skrifið Scheme segðir (mega vera Scheme föll) sem jafngilda þeim. Athugið að í λ-reikningi merkir segð xy fallið x beitt á viðfang y, ekki x margfaldað með y. Í Scheme myndum við skrifa `(x y)` til að fá þessa merkingu. Hins vegar leyfum við okkur í λ-reikningi að nota millitáknun fyrir reikniaðgerðir og þess vegna samsvarar λ-segðin $x * y$ Scheme segðinni `(* x y)`. Í λ-reikningi má að ósekju bæta við svigum án þess að merking breytist, sé það gert á réttan hátt, en í Scheme má hvorki bæta við svigum né fækka þeim án þess að merkingin breytist.

  Write Scheme expressions (may be Scheme functions) that are equivalent to the expressions. Note that in λ-calculus an expression xy means that the function x is applied to argument y, not x multiplied by y. In Scheme we would write `(x y)` to get the same meaning. On the other hand we allow ourselves in λ-calculus to use infix notation for arithmetic operations and therefore the λ-expression $x*y$ corresponds to the Scheme expression `(* x y)`. In λ-calculus we may add parentheses without changing the meaning of the expression, if it is done correctly, but in Scheme we can neither add nor remove parentheses without changing the meaning.

- Ef segðin skilar einföldu gildi (t.d. tölu) skal tiltaka hvert gildið er.

If the expression returns a simple value (e.g. a number) you should specify that value.

- Ef segðin skilar falli sýnið þá, ef hægt er, hvernig nota má fallið í segð sem skilar einföldu gildi.

  If the expression returns a function then show how you might use the function in an expression that returns a simple value.

- Tiltakið hvaða breytur eru frjálsar í hverri segð (ef einhver er). Athugið að hér er spurt um hvort breytan er frjáls í heildarsegðinni, ekki aðeins einhverri undirsegð.

  Show which variables are free in each expression (if any). Note that we are talking about variables that are free in the total expression, not variables that may be free in some subexpression.

- Endurskrifið einnig λ-segðina og skiptið um breytunöfn þar sem það er hægt *án þess að merking hennar breytist* og notið breytunöfn a, b, o.s.frv. í stað x, y o.s.frv.

  Also rewrite the λ-expression and replace variable names with new ones where it is possible *without changing the meaning of the expression* and use variable names a, b, etc. instead of x, y, etc.

Athugið að hér erum við að nota smá viðbætur við venjulegan λ-reikning, sem eru ansi hefðbundnar. Við leyfum okkur t.d. að skrifa $x + y$ og ætlumst til að segðin $5 + 3$ sé umrituð í segðina 8 ef sá möguleiki verður til staðar.

Note that here we are using a slight extension of regular λ-calculus, that is quite conventional. We allow ourselves, for example, to write $x + y$ and we assume that the expression $5 + 3$ may be rewritten to 8 if the opportunity arises.

1. $\lambda x. \left(\lambda y. \left(x + y\right) / y\right)$
2. $\left(\left(\lambda x. \left(\lambda y. \left(x + y\right) / y\right)\right) 3\right) 6$
3. $\left(\left(\lambda x. \left(\lambda y. \left(x \left(xy\right)\right)\right)\right) \left(\lambda x. x^2\right)\right) 3$

Einstaklingsverkefni — Individual Assignments

Íhugið eftirfarandi λ-segðir. Consider the following λ-expressions.

- Skrifið Scheme segðir (mega vera Scheme föll) sem jafngilda þeim. Athugið að í λ-reikningi merkir segð xy fallið x beitt á viðfang y, ekki x margfaldað með y. Í Scheme myndum við skrifa `(x y)` til að fá þessa merkingu. Hins vegar leyfum við okkur í λ-reikningi að nota millitáknun fyrir reikniaðgerðir og þess vegna samsvarar λ-segðin $x * y$ Scheme segðinni `(* x y)`. Í λ-reikningi má að ósekju bæta við svigum án þess að merking breytist, sé það gert á réttan hátt, en í Scheme má hvorki bæta við svigum né fækka þeim án þess að merkingin breytist.

Write Scheme expressions (may be Scheme functions) that are equivalent to the expressions. Note that in λ-calculus an expression xy means that the function x is applied to argument y, not x multiplied by y. In Scheme we would write `(x y)` to get the same meaning. On the other hand we allow ourselves in λ-calculus to use infix notation for arithmetic operations and therefore the λ-expression $x*y$ corresponds to the Scheme expression `(* x y)`. In λ-calculus we may add parentheses without changing the meaning of the expression, if it is done correctly, but in Scheme we can neither add nor remove parentheses without changing the meaning.

- Ef segðin skilar einföldu gildi (t.d. tölu) skal tiltaka hvert gildið er.

  If the expression returns a simple value (e.g. a number) you should specify that value.

- Ef segðin skilar falli sýnið þá, ef hægt er, hvernig nota má fallið í segð sem skilar einföldu gildi.

  If the expression returns a function then show how you might use the function in an expression that returns a simple value.

- Tiltakið hvaða breytur eru frjálsar í hverri segð (ef einhver er). Athugið að hér er spurt um hvort breytan er frjáls í heildarsegðinni, ekki aðeins einhverri undirsegð.

  Show which variables are free in each expression (if any). Note that we are talking about variables that are free in the total expression, not variables that may be free in some subexpression.

- Endurskrifið einnig λ-segðina og skiptið um breytunöfn þar sem það er hægt *án þess að merking hennar breytist* og notið breytunöfn a, b, o.s.frv. í stað x, y o.s.frv.

  Also rewrite the λ-expression and replace variable names with new ones where it is possible *without changing the meaning of the expression* and use variable names a, b, etc. instead of x, y, etc.

Athugið að hér erum við að nota smá viðbætur við venjulegan λ-reikning, sem eru ansi hefðbundnar. Við leyfum okkur t.d. að skrifa $x + y$ og ætlumst til að segðin $5 + 3$ sé umrituð í segðina 8 ef sá möguleiki verður til staðar.

Note that here we are using a slight extension of regular λ-calculus, that is quite conventional. We allow ourselves, for example, to write $x + y$ and we assume that the expression $5 + 3$ may be rewritten to 8 if the opportunity arises.

1. $\lambda x. ((x + z) / z)$
2. $\lambda x. (\lambda y. (\lambda z.x (y (yz))))$