**ZiriTadi koncefciebi rekursiebze**

mravali funqcia savsebiT bunebrivad gani­sazRvreba sxva funqciaTa gamoyenebiT. magaliTad, SesaZlebelia ganvsazRvroT funqcia, romelic gvibrunebs arauaryofiTi mTeli ricxvis faqtorials sabibli­oTeko funqciebis gamoyenebiT erTsa da mocemul mniSvnelobas Soris mo­Tav­sebul ricxvTa namravlis gamosaTvlelad:

*factorial* :: *Int → Int*

*factorial n* = *product* [1*..n* ]

haskelSi daSvebulia agreTve funqciaTa gansazRvra sakuTari Tavis gamoyenebiT. aseT SemTxvevaSi funqciebs *rekursiuls* uwodeben. magaliTad, am gziT SeiZleba ganisazRvros *factorial* funqciac:

*factorial* 0 = 1

*factorial* (*n* + 1) = (*n* + 1)∗ *factorial n*

pirveli gantoleba gveubneba, rom nulis faqtoriali erTia da am gantolebas *sabazo* (*sayrdeni*) *SemTxveva* (*gamosaxuleba*) ewodeba. meore gantoleba amtkicebs, rom nebismieri mkacrad dadebiTi mTeli ricxvis faqtoriali warmoadgens am ricxvisa da misi winamavali ricxvis faqtorialis namravls. mas *rekursiuli SemTxveva* (*gamosaxuleba*) ewodeba.

magaliTad, Semdegi fragmenti gviCvenebs, Tu rogor xdeba samis faqtorialis gamoTvla am gansazRvrebis gamoyenebiT:

*factorial* 3

= { *factorial* funqciis gamoyeneba}

3 ∗ *factorial* 2

= { *factorial* funqciis gamoyeneba}

3 ∗ (2 ∗ *factorial* 1)

= { *factorial* funqciis gamoyeneba}

3 ∗ (2 ∗(1 ∗ *factorial* 0))

= { *factorial* funqciis gamoyeneba}

3 ∗ (2 ∗(1 ∗1))

= {∗ operaciis gamoyeneba}

6

yuradReba miaqcieT im garemoebas, rom, Tumca *factorial* funqcia sakuTari Ta­vis saSualebiT ganisazRvreba, igi usasrulo cikls ar warmoadgens. sa­xel­dobr, *factorial* funqciis yoveli gamoyeneba amcirebs mTelricxva ar­guments erTiT, vidre igi saboloo jamSi nulis toli ar gaxdeba. aq re­kursia wydeba da gamravlebis operacia sruldeba. nulis faqtoriali, rom­liTac erTiani gvibrundeba, savsebiT adekvaturia am viTarebaSi, rad­gan gamravlebisas erTiani igiveobas unarCunebs gamosaxulebas. sxvanairad rom vTqvaT, 1 ∗*x*= *x* da *x*∗1 = *x* nebismieri mTeli *x* ricxvisaTvis.

*factorial* funqciis SemTxvevaSi sawyisi gansazRvreba sabiblioTeko funqciebis saSualebiT ufro martivia, vidre rekursiis gamoyenebiT. magram, rogorc amas wignis darCenil nawilSi davinaxavT, mravali funqcia martivad da bunebrivad swored rekursiis gamoyenebiT ganisazRvreba. magaliTad, mravali sabiblioTeko funqcia haskelSi ganisazRvreba rekursiiT. garda amisa, rogorc me-13 TavSi vnaxavT, funqciaTa gansazRvra rekursiiT maTi Tvisebebis damtkicebis saSualebas iZleva maTematikuri induqciis mZlavri meTodis gamoyenebiT.

mTel ricxvebze rekursiis kidev erT magaliTad ganvixiloT zemoT gamoyenebuli gamravlebis ∗ operatori. efeqturobis mosazrebebidan gamomdinare, haskelSi es operatori gaTvaliswinebulia rogorc primitivi \_ daprogramebis enaSi programaTa Sesrulebis siCqaris gazrdisaTvis CaSenebuli operatori. magram arauaryofiTi mTeli ricxvebisaTvis igi SeiZleba iyos gansazRvruli aseve rekursiiTac Tavisi ori argumentidan nebismier erTze, vTqvaT meoreze:

(∗) :: *Int → Int → Int*

*m* ∗0 = 0

*m* ∗(*n* + 1) = *m* + (*m* ∗ *n*)

magaliTad:

4 ∗3

= {∗opratoris gamoyeneba }

4 + (4 ∗2)

= {∗opratoris gamoyeneba }

4 + (4 + (4 ∗1))

= {∗opratoris gamoyeneba }

4 + (4 + (4 + (4 ∗ 0)))

= { ∗opratoris gamoyeneba }

4 + (4 + (4 + 0))

= {+ opratoris gamoyeneba }

12

amrigad, aq rekursiuli gansazRvreba ∗ operatorisaTvis formalurad gamoxatavs im ideas, rom gamravleba daiyvaneba iteraciul (mravaljerad) Sekrebamde.

**rekursia siebze**

rekursia ar Semoifargleba funqciebiT mTel ricxvebze, igi SeiZleba aseve gamoyenebuli iqnes funqciaTa gansazRvrisaTvis siebze. magaliTad, *product* sabiblioTeko funqcia, romelic wina punqtSi vixmareT, Semdegi saxiT SeiZleba ganisazRvros:

*product* :: *Num a* ⇒[a] *→ a*

*product* [] = 1

*product* (*n* : *ns*) = *n* ∗ *product ns*

pirveli gantoleba amtkicebs, rom carieli siis namravli erTiania, rac savsebiT adekvaturia, radgan erTiani gamravlebis operaciaSi unarCunebs gamosaxulebas igiveobas. meore gantoleba ki gveubneba, rom nebismieri aracarieli siis namravli miiReba pirveli ricxvisa da ricxvTa darCenili siis namravlis erTmaneTze gamravlebis operaciiT. magaliTad:

*product* [2*,* 3*,* 4]

= { *product* funqciis gamoyeneba }

2 *∗ product* [3*,* 4]

= { *product* funqciis gamoyeneba }

2 ∗(3 ∗ *product* [4])

= { *product* funqciis gamoyeneba }

2 ∗(3 ∗ (4 ∗ *product* []))

= { *product* funqciis gamoyeneba }

2 ∗(3 ∗(4 ∗1))

= {∗operatoris gamoyeneba }

gavixsenoT, rom realurad sias haskelSi aqvs erTi elementis logikuri struqtura, romelic amave dros cons operators iyenebs. maSasadame, [2*,* 3*,* 4] Canaweri mxolod abreviaturaa 2 : (3 : (4 : [])) gamosaxulebisaTvis da meti araferi. ganvixiloT siebze rekursiis kidev erTi martivi magaliTi, risTvisac mivmarToT *length* sabiblioTeko funqcias, romelic SeiZleba ganisazRvros rekursiis amave Sablonis gamoyenebiT *product* funqciis msgavsad:

*length* :: [*a*] *→ Int*

*length* [] = 0

*length* ( \_ : *xs*) = 1 + *length xs*

maSasadame, carieli siis sigrZe nulia, xolo nebismieri aracarieli siis sigrZe misi kudis sigrZis momdevno mniSvnelobaa. yuradReba miaqcieT ( \_ ) Casmis Sablonis gamoyenebas rekursiul gamosaxulebaSi. es Sabloni asaxavs im faqts, rom siis sigrZe damokidebuli ar aris misi elementebis mniSvnelobaze.

axla ganvixiloT sabiblioTeko funqcia, romelic siis Sebrunebas, Seqcevas (reverss) axorcielebs. rekursiis saSualebiT es funqcia SeiZleba Semdegnairad ganvsazRvroT:

*reverse* :: [*a*] *→* [*a*]

*reverse* [] = []

*reverse* (*x* : *xs*) = *reverse xs* ++ [*x*]

maSasadame, carieli siis reversi kvlav carieli siaa, xolo nebismieri aracarieli siis reversi xorcieldeba mibmiT misi kudis reversTan erTelementiani siis, romelic sawyisi siis Tavs warmoadgens. magaliTad:

*reverse* [1*,* 2*,* 3]

= { *reverse* funqciis gamoyeneba }

*reverse* [2*,* 3] ++ [1]

= { *reverse* funqciis gamoyeneba }

(*reverse* [3] ++ [2])++ [1]

= { *reverse* funqciis gamoyeneba }

((*reverse* []++ [3])++ [2])++ [1]

= { *reverse* funqciis gamoyeneba }

(([] ++ [3])++ [2]) ++ [1]

= {++ operatoris gamoyeneba }

[3*,* 2*,* 1]

Tavis mxriv, damatebis ++ operatori, romelic *reverse* funqciis zemoT mocemul gansazRvrebaSi gamoiyeneba, Tavad SeiZleba iqnes aRwerili rekursiiT Tavis pirvel argumentze:

(++) :: [*a*] *→* [*a*] *→* [*a*]

[]++ *ys* = *ys*

(*x* : *xs*) ++ *ys* = *x* : (*xs* ++ *ys*)

magaliTad:

[1*,* 2*,* 3] ++ [4*,* 5]

= {++ operatoris gamoyeneba }

1 : ([2*,* 3] ++ [4*,* 5])

= { ++ operatoris gamoyeneba }

1 : (2 : ([3]++ [4*,* 5]))

= {++ operatoris gamoyeneba }

1 : (2 : (3 : ([] ++ [4*,* 5])))

= {++ operatoris gamoyeneba }

1 : (2 : (3 : [4*,* 5]))

= {Cawera siis saxiT }

[1*,* 2*,* 3*,* 4*,* 5]

amrigad, rekursiuli gansazRvreba ++ operatorisaTvis formalurad asaxavs im ideas, rom ori siis gadabma SeiZleba pirveli siidan elementebis piris (aslis) gadaRebiT, vidre es sia ar amoiwureba, ris Semdeg meore sia daemateba am asls boloSi.

ganvixiloT kidev ori magaliTi, romelic exeba rekursias daxarisxebul siebze. upirveles yovlisa, ganvixiloT funqcia, romelic daxarisxebul siaSi axorcielebs nebismieri mowesrigebuli tipis axali elementis Casmas kidev erTi axali daxarisxebuli siis misaRebad. es funqcia Semdegi saxiT SeiZleba ganisazRvros:

*insert* :: *Ord a* ⇒ *a →* [*a*] *→* [*a*]

*insert x* [] = [*x*]

*insert x* (*y* : *ys*) *| x* ≤ *y* = *x* : *y* : *ys*

*| otherwise* = *y* : *insert x ys*

maSasadame, axali elementis Casma cariel siaSi iZleva erTelementian sias, maSin roca aracarieli siisaTvis Sedegi damokidebulia axali *x* elementisa da siis *y* Tavis urTierTganlagebaze. saxeldobr, Tu *x* ≤ *y*, maSin axali *x* elementi mxolod Tavsdeba siis dasawyisSi da meti araferi, winaaRmdeg SemTxvevaSi *y* Tavi saboloo siis pirveli elementi xdeba da Semdeg iwyeben axali elementis Casmas mocemuli siis kudSi. magaliTad:

*insert* 3 [1*,* 2*,* 4*,* 5]

= { *insert* funqciis gamoyeneba }

1 : *insert* 3 [2*,* 4*,* 5]

= { *insert* funqciis gamoyeneba }

1 : 2 : *insert* 3 [4*,* 5]

= { *insert* funqciis gamoyeneba }

1 : 2 : 3 : [4*,* 5]

= {Cawera siis saxiT }

[1*,* 2*,* 3*,* 4*,* 5]

*insert* funqciis gamoyenebiT SesaZlebelia axali funqciis gansazRvra, romelic axor­cielebs *daxarisxebas CasmiT*[[1]](#footnote-1) (ingl. *insertion sort*). saxeldobr, am gansa­zRvre­baSi unda iqnes gaTvaliswinebuli, rom carieli sia ukve daxarisxebulia, xolo nebismieri aracarieli siis daxarisxeba xdeba misi Tavis CasmiT kudis daxarisxebis Sedegad miRebul siaSi:

*isort* :: *Ord a* ⇒ [a] *→* [a]

*isort* [] = []

*isort* (*x* : *xs*) = *insert x* (*isort xs*)

magaliTad:

*isort* [3*,* 2*,* 1*,* 4]

= { *isort* funqciis gamoyeneba }

*insert* 3 (*insert* 2 (*insert* 1 (*insert* 4 [])))

= { *insert* funqciis gamoyeneba }

*insert* 3 (*insert* 2 (*insert* 1 [4]))

= { *insert* funqciis gamoyeneba }

*insert* 3 (*insert* 2 [1*,* 4])

= { *insert* funqciis gamoyeneba }

*insert* 3 [1*,* 2*,* 4]

= { *insert* funqciis gamoyeneba }

[1*,* 2*,* 3*,* 4]

**mravalargumentiani funqciebi**

amasTan erTad, mravalargumentiani funqciebi aseve SeiZleba ganisaz­R­vros rekursiis gamoyenebiT or an met argumentze. magaliTad, sabib­li­o­Te­ko *zip* funqcia, romelic iRebs Sesasvlelze or sias da wyvilebis Se­q­mnil sias gvibrunebs, Semdegi saxiT ganisazRvreba:

*zip* :: [*a*] *→* [*b*] *→* [(*a, b*)]

*zip* [] \_ = []

*zip \_* [] = []

*zip* (*x* : *xs*) (*y* : *ys*) = (*x , y*) : *zip xs ys*

magaliTad:

*zip* ['a'*,* 'b'*,* 'c'] [1*,* 2*,* 3*,* 4]

= { *zip* funqciis gamoyeneba }

('a'*,* 1) : *zip* ['b'*,* 'c'] [2*,* 3*,* 4]

= { *zip* funqciis gamoyeneba }

('a'*,* 1) : ('b'*,* 2) : *zip* ['c'] [3*,* 4]

= { *zip* funqciis gamoyeneba }

('a'*,* 1) : ('b'*,* 2) : ('c'*,* 3) : *zip* [] [4]

= { *zip* funqciis gamoyeneba }

('a'*,* 1) : ('b'*,* 2) : ('c'*,* 3) : []

= {Cawera siis saxiT }

[('a'*,* 1)*,* ('b'*,* 2)*,* ('c'*,* 3)]

yuradReba miaqcieT, rom *zip* funqciis gansazRvrebaSi saWiroa ori saba­zo gamosaxuleba, vinaidan argumentTa ori siidan nebismieri SeiZleba aRmoCndes carieli.

ganvixiloT ramdenime argumentze rekursiis kidev erTi magaliTi. saxel­dobr, sabiblioTeko *drop* funqcia, romelic siaSi anadgurebs elemen­te­bis mocemul raodenobas am siis dasawyisidan, Semdegi saxiT SeiZleba ga­ni­sa­zRvros:

*drop* :: *Int →* [*a*] *→* [*a*]

*drop* 0 *xs* = *xs*

*drop* (*n* + 1) [] = []

*drop* (*n* + 1) ( \_ : *xs*) = *drop n xs*

aq kvlav ori sabazo (sayrdeni, sayrdnobi) gamosaxulebaa saWiro: erTi \_ nulovani raodeno­bis elementTa gasanadgureblad siaSi, xolo meore \_ cariel siaSi erTi an ramdenime elementis ganadgurebis mcdelobisas.

**mravaljeradi rekursia**

funqciebi aseve SeiZleba ganisazRvros *mravaljeradi rekursiiT*, romel­Sic funqcia Tavis sakuTar gansazRvrebas araerTxel iyenebs. magaliTad, gavi­xsenoT fibonaCis ricxvTa 0*,* 1*,* 1*,* 2*,* 3*,* 5*,* 8*,* 13*, . . .* mimdevroba, ro­mel­Sic pirveli da meore ricxvia 0 da 1 Sesabamisad, xolo yoveli momdev­no ricxvi wina oris jams warmoadgens. haskelSi funqcia, romelic anga­ri­Sobs fibonaCis me-*n* ricxvs nebismieri mTeli *n* ≥ 0 ricxvisaTvis, Sem­deg­nairad SeiZleba ganisazRvros orjeradi rekursiis gamoyenebiT:

*fibonacci* :: *Int → Int*

*fibonacci* 0 = 0

*fibonacci*1 = 1

*fibonacci* (*n* + 2) = *fibonacci n* + *fibonacci* (*n* + 1)

kidev erT msgavs magaliTs mivmarToT. saxeldobr, gavixsenoT, rom pir­vel TavSi naCvenebi iyo siis swarafi daxarisxebis (mowesrigebis) quicksort saxelwodebiT cnobi­li meTodis ganxorcieleba. Sesabamisi funqciis gansazRvra Semdegi sa­xiT SeiZleba:

*qsort* :: *Ord a* ⇒[*a*] *→* [*a*]

*qsort* [] = []

*qsort* (*x* : *xs*) = *qsort smaller* ++ [*x*] ++ *qsort larger*

**where**

*smaller* = [*a | a ← xs, a* ≤ *x*]

*larger* = [*b | b ← xs, b > x*]

maSasadame, carieli sia ukve daxarisxebulia, xolo nebismieri aracarieli sia SeiZleba daxarisxdes, Tu mis Tavs movaTavsebT or sias Soris. isini warmoadgens mocemuli aracarieli siis kudis im elementebis daxarisxebis Sedegs, romlebic an naklebia am Tavze, an masze metia Sesabamisad.

**urTierTrekursia**

funqciebis gansazRvra aseve SeiZleba urTierTrekursiiT (ingl. *mutual re­cur­sion*), roca ori an meti funqcia mTlianad gansazRvrulia erT­ma­ne­Tis saSualebiT. magaliTad, ganvixiloT *even* (luwi ricxvi) da *odd* (ken­ti ricxvi) sabiblioTeko funqciebi. efeqturobis gazrdis mizniT es funq­ciebi, Cveulebriv, ganisazRvreba orze gayofis Sedegad miRebuli naS­Tis gamoyenebiT. magram arauaryofiTi mTeli ricxvebisaTvis maTi gan­sa­zRvra urTierTrekursiis saSualebiTac SeiZleba:

*even* :: *Int → Bool*

*even* 0 = *True*

*even* (*n* + 1) = *odd n*

*odd* :: *Int → Bool*

*odd* 0 = *False*

*odd* (*n* + 1) = *even n*

maSasadame, nuli \_ mTeli ricxvia, xolo nebismieri mkacrad dadebiTi ricxvi luwia, Tu misi wina ricxvi kentia, da nebismieri mkacrad dadebiTi ricxvi kentia, Tu misi wina ricxvi luwia. magaliTad:

*even* 4

= { *even* funqciis gamoyeneba }

*odd* 3

= { *odd* funqciis gamoyeneba }

*even* 2

= { *even* funqciis gamoyeneba }

*odd* 1

= { *odd* funqciis gamoyeneba }

*even* 0

= { *even* funqciis gamoyeneba }

*True*

amis msgavsad, funqciebi, romlebic irCevs elementebs siidan yvela luw da kent poziciaSi (am poziciis aTvlisas nulidan), SeiZleba Semdegi saxiT ganisazRvros Sesaamisad:

*evens* :: [*a*] *→* [*a*]

*evens* [] = []

*evens* (*x* : *xs*) = *x* : *odds xs*

*odds* :: [*a*] *→* [*a*]

*odds* [] = []

*odds* ( : *xs*) = *evens xs*

magaliTad:

*evens* "abcde"

= { *evens* funqciis gamoyeneba }

'a' : *odds* "bcde"

= { *odds* funqciis gamoyeneba }

'a' : *evens* "cde"

= { *evens* funqciis gamoyeneba }

'a' : ’c’ : *odds* "de"

= { *odds* funqciis gamoyeneba }

'a' : 'c' : *evens* "e"

= { *evens* funqciis gamoyeneba }

'a' : 'c' : 'e' : *odds* []

= { *odds* funqciis gamoyeneba }

'a' : 'c' : 'e' : []

= {Cawera siis saxiT }

"ace"

gavixsenoT, rom striqonebi haskelSi realurad agebulia rogorc simboloTa siebi. amrigad, "abcde" Canaweri mxolod abreviaturaa ['a'*,* 'b'*,* 'c'*,* 'd'*,* 'e'] gamosaxulebisaTvis da meti araferi.

1. xeliT daxarisxebis martivi da Zalian araefeqturi meTodi, roca monacemTa morigi elementi Tavsdeba siis saWiro adgilze siis arsebul elementebTan Sedarebis Semdeg. [↑](#footnote-ref-1)