

Uvod u interaktivno dokazivanje teorema

Vežbe 11

Zadatak 1 *Tip: 'a drvo*

Definisati algebarski tip *'a drvo* koji predstavlja binarno drvo.

datatype *'a drvo* = *undef*

Definisati funkciju *zbir* :: *nat drvo* \Rightarrow *nat* primitivnom rekurzijom koja računa zbir elemenata drveta tipa *nat drvo*. Da li je moguće definisati ovu funkciju nad tipom *'a drvo*?

Definisati bilo koju instancu *test-drvo* tipa *nat drvo*. Proveriti da li funkcija *zbir* daje dobar rezultat kada se primeni na *test-drvo*.

Definisati funkciju *sadrzi* :: *'a drvo* \Rightarrow *'a* \Rightarrow *bool* primitivnom rekurzijom koja proverava da li se dati element nalazi u drvetu. Takođe, testirati funkciju nad instancom *test-drvo*.

Definisati funkciju *skup* :: *'a drvo* \Rightarrow *'a set* primitivnom rekurzijom koja proverava da li se dati element nalazi u drvetu. Takođe, testirati funkciju nad instancom *test-drvo*.

Pronaći vezu između funkcija *skup* i *sadrzi*. Formulirati i dokazati tu lemu.

Zadatak 2 *Obilazak stabla*

Definisati funkciju *infiks* koja vraća listu čvorova stabla u infiksnom poretku.

Pokazati korektnost ove funkcije. Dve invarijante:

1. *Skup elemenata infiksnog obilaska drveta i skup elemenata drveta ostaju isti.*
2. *Multiskup elemenata infiksnog obilaska drveta i skupa elemenata drveta ostaju isti.*

Savet: Tip multiskupa: *'a multiset*, prazan multiskup se definiše kao $\{\#\}$, multiskup sa jednim elementom $\{\#x\#\}$, unija multiskupova je operator $+$.

Definisati efikasnu implementaciju infiksnog obilaska drveta *infiks-opt* i pokazati da je ekvivalentna funkciju *infiks*.

definition *infiks-opt* :: *'a drvo* \Rightarrow *'a list* **where**
infiks-opt *xs* = *undefined*

Zadatak 3 *Binarno pretraživačko stablo.*

Definisati predikat *sortirano* nad binarnim stablom tipa (*'a::linorder*) *drvo* koji ukazuje na to da li je stablo pretraživačko ili nije. Definisati instancu *test-drvo-sortirano* nad tipom *nat drvo* koja predstavlja binarno pretraživačko stablo. Testirati funkciju *sortirano* nad instancom *test-drvo* i *test-drvo-sortirano*. Zapisati i dokazati vezu između funkcije *sortirano* i *infiks*.

definition *test-drvo-sortirano* :: *nat drvo* **where**
test-drvo-sortirano = *undefined*

Primitivnom rekurzijom definisati funkciju $ubaci :: 'a::linorder \Rightarrow 'a\ drvo \Rightarrow 'a\ drvo$ koja ubacuju element u binarno pretraživačko drvo.

Pokazati da važe invarijante:

1. *Element će se nalaziti u drvetu nakon što se ubaci.*
2. *Skup elemenata drveta nakon ubacivanja elementa se proširuje za taj element.*
3. *Multiskup elemenata drveta nakon ubacivanja elementa se proširuje za taj element.*
4. *Zbir elemenata drveta nakon ubacivanja elementa se povećava za njegovu vrednost.*
5. *Nakon ubacivanja elementa u pretraživačko drvo, drvo ostaje pretraživačko.*

Definisati funkciju $listaUDrvo :: ('a::linorder)\ list \Rightarrow 'a\ drvo$ koja od liste elemenata gradi binarno pretraživačko drvo.

Pokazati sledeće osobine funkcije $listaUDrvo$:

1. *$listaUDrvo$ održava skup elemenata.*
2. *$listaUDrvo$ održava multiskup elemenata.*
3. *$listaUDrvo$ gradi binarno pretraživačko drvo.*

Definisati funkciju koja sortira elemente liste pomoću stabla:

definition $sortiraj :: nat\ list \Rightarrow nat\ list$ **where**
 $sortiraj\ xs = undefined$

Pokazati korektnost ove funkcije

1. *Nakon primene funkcije $lista$ je sortirana.*
2. *Skup elemenata sortirane liste i početne liste ostaje isti.*
3. *Multiskup elemenata sortirane liste i početne liste ostaje isti.*