Programazioaren Metodologia

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Gradua
Bilboko Ingeniaritza Eskola (UPV/EHU)
Lengoaia eta Sistema Informatikoak Saila
1. maila
2019-2020 ikasturtea
31 taldea
4. gaia: Programak era formalean eratortzeko metodoa
1,5 puntu

Lana

Enuntziatua

2020-04-20

Aurkibidea

Irudien zerrenda

Taulen zerrenda

****

1 Programa iteratibo bat eratortzea (*1,5 puntu*)

Osoak eta positiboak diren zenbakiz eratuta dagoen $A(1..n)$ bektore ez-hutsa sarrerako datu gisa hartuta, q aldagai boolearrean $A(1..n)$ bektoreko 2 eta n posizioen arteko elementuren bat, elementuari berari dagokion posizioaren anizkoitza al den erabakiko duen programa eratorri behar da. Programa eratortzeko, emandako hasierako eta bukaerako baldintzak (φ eta ψ), INB inbariantea eta E espresioa hartu behar dira kontuan eta Hoare-ren kalkuluko While-aren Erregela eta Esleipenaren Axioma erabili behar dira. Lortutako programak eraginkorra izan beharko du, hau da, uneren batean erantzuna baiezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainerako posizioak aztertu gabe.

1 irudian, eratorri beharreko programaren egitura, φ, ψ, INB eta E-ren definizioa eta φ eta INB formuletan erabilitako predikatuaren definizioa daude.

1 irudian, mod eragilea zatiketa osoaren hodarra adierazteko erabili da. Adibideak: $20\ mod\ 3 = 2$, $18\ mod\ 3 = 0$, $19\ mod\ 3 = 1$. Hiru adibide horietan, div eragilearen bidez adieraziko dugun zatiketa osoak 6 balioa itzuliko luke: $20\ div\ 3 = 6$, $18\ div\ 3 = 6$, $19\ div\ 3 = 6$. Zatiketa osoarentzat beste adibide batzuk: $19\ div\ 2 = 9$; $19\ div\ 3 = 6$; $19\ div\ 4 = 4$; $17\ div\ 3 = 5$; $8\ div\ 12 = 0$.

Eratortze-prozesuan, 2. orrialdean dagoen 1 taulan agertzen diren laburdurak erabiltzea komeniko litzateke. Bestalde, 3. orrialdean dagoen 2 taulan, enuntziatu honetan erabili diren letra grekoak jaso dira. Azkenik, 4. orrialdean dagoen 3 taulan, eratortze-prozesuan kontuan hartu beharreko urratsei edo atalei dagozkien puntuazioak ipini dira.

1 irudian eta 1 taulan agertzen diren zenbakizko elementuen bidez adierazitako balioak zenbaki osoak dira. Beraz, elementu horien bidez adierazitako balioak $\mathbb{Z}$ multzokoak dira. $\mathbb{Z}$ multzoa honako multzo hau da: $\{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots\}$.

Formalki, $\mathbb{Z} = \mathbb{N} \ \cup \ \{-y \mid y \in \mathbb{N} \ \wedge \ y \geq 1\}$. Definizio horretan, $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ zenbaki arrunten multzoa da eta $\cup$ multzoen arteko bilketa adierazteko erabili da. Beraz, $\mathbb{Z}$ multzoa $\mathbb{N}$ eta $\{-y \mid y \in \mathbb{N} \ \wedge \ y \geq 1\}$ multzoen arteko bildura da.

Adibidea. (Eratorri beharreko programarentzat. Programa horren egitura, 1 irudian dago) *Har dezagun honako $A(1..8)$ bektorea:*

A(1..8)	*10*	*9*	*10*	**8**	*17*	**30**	*4*	*2*
	1	*2*	*3*	**4**	*5*	**6**	*7*	*8*

Eratorri beharreko programak, $A(1..8)$-ren balio horientzat $True$ balio boolearra laga beharko luke q aldagaian. Izan ere, $A(1..8)$ bektoreko 4 posizioko elementua (hau da, 8) 4ren (posizioaren) anizkoitza da eta 6 posizioko elementua (hau da, 30) 6ren (posizioaren) anizkoitza da. Beraz, 2 eta 8-ren arteko posizioak kontuan hartuta, gutxienez posizio bateko elementua elementuari berari dagokion posizioaren anizkoitza denez, $True$ erantzuna itzuli beharko da. Eratorri behar den programaren egitura 1 irudian ikus daiteke.

Aldiz, $A(1..8)$ bektorearen balioak beste hauek balira, orduan programak $False$ balio boolearra laga beharko luke q aldagaian. Izan ere, 2 eta 8-ren arteko posizioak kontuan hartuta, elementu bat bera ere ez da dagokion posizioaren anizkoitza:

A(1..8)	*2*	*5*	*2*	*7*	*3*	*39*	*18*	*3*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*

Honako laburdura hauek erabiltzea aholkatzen da:
$\lambda \equiv n \geq 2 \ \wedge \ posit(A(1..n))$ $\gamma(\ell) \equiv (A(\ell)\ mod\ \ell) = 0$ $\mu(\ell) \equiv \forall k((2 \leq k \leq \ell) \ \rightarrow \ ((A(k)\ mod\ k) \neq 0))$

1 taula: Aholkatutako laburdurak.

Eratorri beharreko programaren egitura:
$\{\varphi\}$ Hasieraketak? **while** $\{INB\}$ $\{E\}$ B? **loop**   Aginduak? **end loop;** $\{\psi\}$
φ, INB, E eta ψ-ren definizioak:
$\varphi \equiv \quad n \geq 2 \ \wedge \ posit(A(1..n))$ $INB \equiv \quad n \geq 2 \ \wedge \ posit(A(1..n)) \ \wedge \ (2 \leq i \leq n) \wedge$ $(q \leftrightarrow ((A(i) \ mod \ i) = 0)) \wedge$ $\forall k((2 \leq k \leq i-1) \ \rightarrow \ ((A(k) \ mod \ k) \neq 0))$ $E = \quad n - i$ $\psi \equiv \quad q \leftrightarrow \exists k((2 \leq k \leq n) \ \wedge \ ((A(k) \ mod \ k) = 0))$
Erabilitako predikatuaren definizioa:
$posit(H(1..r)) \equiv \forall k((1 \leq k \leq r) \ \rightarrow \ (H(k) \geq 1))$

1 irudia: Eratorri beharreko programaren egitura, φ, INB, E eta ψ-ren definizioak eta erabilitako predikatuaren definizioa.

Enuntziatuan erabili diren letra grekoak:
φ : fi ψ : psi γ : gamma μ : mu λ : lambda

2 taula: Enuntziatuan erabili diren letra grekoen izenak.

Puntuazioa:

(a) While-aren aurreko hasieraketak kalkulatzea: 0,250
(b) While-aren baldintza (B) kalkulatzea: 0,380
    (b.1) $\neg B$ eta B formulatzea: 0,125
    (b.2) While-aren erregelako (II) puntua egiaztatzea: 0,030
    (b.3) While-aren erregelako (IV) puntua egiaztatzea: 0,200
    (b.4) While-aren erregelako (V) puntua egiaztatzea: 0,025
(c) While-aren barruko aginduak kalkulatzea: 0,850
    (c.1) While-aren erregelako (III) puntuari lotutako garapena: 0,550
    (c.2) While-aren erregelako (VI) puntuari lotutako garapena: 0,300
(d) d) Bukaeran programa osoa idaztea: 0,020
 	
■ Inplikazio bat zergatik betetzen den ez bada azaltzen, zero kontatuko da. Hau da, inplikazio bat betetzen dela esateak zergatik betetzen den azaldu gabe, zero balio du.
 	
■ Ariketa hau gainditzeko, *(a)*, *(b)* eta *(c)* ataletan, atal horietako puntuazioaren erdia lortu beharko da.

3 taula: Puntuazioa atalka.