Programozási módszertan

1. Alapfogalmak

Feldhoffer Gergely

2012

Féléves tananyag terve

- Program helyességének bizonyítása
  - Reprezentáció
  - Logikai-matematikai eszköztár
  - Programozási tételek bizonyítása
  - Levezetés - visszavezetés
- Programhelyesség bizonyításának alternatív formalizmusai
- Gyakorlatban használható eszközök
  - debug technikák (gdb használat)
  - fehér doboz tesztelési eszközök (coverage)
  - szimulációs segédeszközök (valgrind)

A tananyag elméleti részét Fóthi Ákos dolgozta ki, a diákban nagyban felhasználom Fehér Tamás jegyzetét.

Jegyszerzés

A félév során a gyakorlati részből két ZH-t kell megírni, egy elméleti papírost, és egy géptermit. Az itt kapott értékelés az alapja az aláírásnak. A félév végén vizsgázni kell. A tantárgyra kapott jegy a vizsgán kapott jegy, a ZH eredményeket szubjektíven veszem figyelembe.

Jegyszerzés

A félév során a gyakorlati részből két ZH-t kell megírni, egy elméleti papírost, és egy géptermit. Az itt kapott értékelés az alapja az aláírásnak. A félév végén vizsgázni kell. A tantárgyra kapott jegy a vizsgán kapott jegy, a ZH eredményeket szubjektíven veszem figyelembe.
Amennyiben a papíros ZH eredménye elmarad a várakozástól, a pluszmínusz íratás jogát a félév második felére fenntartom, reméljük erre nem kerül sor.

Mi az a program?

A továbbiakban adunk egy olyan definíciót a programra, ami nem tökéletesen fedi az intuitív program fogalmat, valahol az algoritmus és a program általánosan használt fogalmai közé pozícionálva

Az *algoritmus* megoldási menetet rögzít, függetlenül a reprezentációtól, implementációtól.
A *program* általában konkrét algoritmus konkrét reprezentációjának konkrét implementációja.

Egy matematikai algoritmus például értékes lehet akkor is, ha ábrázolhatatlanul nagy számokra lehet csak levezetni az érdekes esetben – vagy egy végtelen sor n-edik részösszegének pontos értéke is nehezen ábrázolható teljes pontossággal.

Az általunk használt program definíciót úgy terjesztjük ki a hagyományos program fogalomból, hogy megengedünk a valóságban implementálhatatlan típusokat, mint pl. $\mathbb{N}$ vagy $\mathbb{R}$

Állapottér

A program viselkedését az azt futtató gép belső állapota szabályozza. Ezen állapot több, egyesével típushoz rendelhető változó együttese. Például egy számítógép processzorának regisztereinek értékei, a memóriájának összes értéke, a háttértárak és inputeszközökből származó adat együtt lefedi azt a kört, ami egy program következő lépését meghatározza.

Mivel ez kezelhetetlenül nagy dimenzionalitás, illetve a típusosság előnyeit nem tartalmazza, a bizonyítás során megelégszünk a program által használt változók gyűjteményével. Ezek direktszorzata az állapottér:

Def (Állapottér)

Legyenek $A_1, A_2, \ldots, A_n$ tetszőleges véges, vagy megszámlálható nem üres halmazok. Ekkor az $A = A_1 \times A_2 \times \cdots \times A_n$ halmazt **állapottér***nek, az A_i halmazokat pedig* **típusértékhalmaz***oknak nevezzük.*

A feladat fogalma

A feladat az állapottér valamely pontjáról (input) eljutni a megfelelő pontba (eredmény), vagyis

Def (Feladat)

Feladat*nak nevezzük az* $F \subseteq A \times A$ *relációt.*

Vegyük észre, hogy

- nem kötjük ki, hogy minden kezdeti állapotról rendelkezzen a feladat
- nem csak egyetlen jó megoldást rendelhet a feladat az inputhoz

A feladat fogalma

Például az összeadás feladata a következő:

$A = \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$
$F = \{((a, b, c), (d, e, f))|f = a + b\}$

Amennyiben az összeadástól elvárjuk, hogy megtartsa a bemenő paraméterek értékét, akkor
$F = \{((a, b, c), (d, e, f))|f = a + b \wedge d = a \wedge e = b\}$

Az állapottérbeli pontok sorozatai

Egy program az azt futtató gép memóriáját folyamatosan változtatja. Ez az állapottér sokdimenziós terében egy pontsorozatként fogható fel: minden időpillanatban az állapottért egy adott pontjában vagyunk, és ahogy az idő telik, megváltozik valamelyik változónk értéke, ugrunk a következő állapotba.

Jelöljük A^*-gal az A halmaz elemeiből képzett véges sorozatokat, A^∞-vel a végtelen sorozatokat, A^{**} legyen $A^* \cup A^\infty$.

Jelölje $red(\alpha)$ egy $\alpha \in A^{**}$ sorozat **redukált**ját, ha $\beta = red(\alpha)$ úgy kapható α sorozatból, hogy minden véges hosszú azonos állapotokból álló részsorozatot egyetlen elemre rövidítünk.

Jelölje $\tau(\alpha)$ egy $\alpha \in A^*$ véges sorozat utolsó elemét, azaz ha $\alpha =< a_1 a_2 \ldots a_n >$ akkor $\tau(\alpha) = a_n$.

A program fogalma

Def (Program)

Program*nak nevezzük az* $S \subseteq A \times A^{**}$ *relációt, ha*

1. $\mathcal{D}_S = A$
2. $\forall a \in A : \forall \alpha \in S(a) : \alpha_1 = a$
3. $\forall \alpha \in \mathcal{R}_S : \alpha = red(\alpha)$

Azaz

1. A program minden lehetséges kezdeti állapotból elindul.
2. Minden a elemhez rendelt α sorozat első eleme az a, kezdeti állapot.
3. A programban minden lépéssorozat redukált

A Program tulajdonságai

Ha egy $\alpha \in \mathcal{R}_S$ elemei között két egymást követő állapot egyforma, akkor garantáltan végtelenszer ugyanaz az állapot fog ismétlődni a redukáltság miatt

Egy $a \in A$ kezdeti állapothoz rendelhet több A^{**}-beli elemet is a program, ilyenkor a program nemdeterminisztikus. Más szavakkal a program **determinisztikus**, ha $\forall a \in A : |S(a)| = 1$.
A nemdeterminisztikus program működését úgy kell tekinteni, hogy a lehetséges utak közül bármelyiket választhatja.

Programfüggvény

Def (Programfüggvény)

A $p(S) \subseteq A \times A$ reláció az $S \subseteq A \times A^{**}$ program **programfüggvénye**, *ha*

1. $\mathcal{D}_{p(S)} = \{a \in A | S(a) \subseteq A^*\}$
2. $p(S)(a) = \{b \in A | \exists \alpha \in S(a) : \tau(\alpha) = b\}$

Azaz

1. A programfüggvény értelmezési tartománya az állapottér azon elemeit tartalmazza, amelyekre a program terminál
2. determinisztikus program esetében a végállapotból álló egy elemű halmazt, nemdeterminisztikus esetben a lehetséges végállapotok halmazát rendeli a kezdeti állapothoz

Megoldás

Def (Megoldás)

Azt modjuk, hogy as S program megoldja az F feladatot, ha

1. $\mathcal{D}_F \subseteq \mathcal{D}_{p(S)}$,
2. $\forall a \in \mathcal{D}_F : p(S)(a) \subseteq F(a)$.

Azaz

1. A feledatban meghatározott pontokból a program terminál
2. A program lefutása után a végállapot csak olyan elem lehessen, amit a feladat hozzárendelt a kezdeti állapothoz.

Megoldás

A program viselkedését nem szabályozza a $\mathcal{D}_F$-en kívüli állapotokra a feladat, ezen pontokban tetszőleges viselkedés mellett megoldhatja a program az F feladatot

Determinisztikus program esetében a szakirodalom néha használja azt a megoldásdefiníciót, ahol a $\forall a \in \mathcal{D}_F : p(S)(a) \in F(a)$ feltétel szerepel.

Alapfogalmak összefoglaló

Az állapottér leírja, hogy milyen lehetséges állapotokat érinthet a program a futása során

A feladat megadja, hogy mely pontokból mik az elfogadható eredmények

A program minden kezdeti állapotból induló, időben állapotok között ugráló folyamat, amit sorozatok formájában kezelünk

Azok ezek a sorozatok véget érnek, ott van eredmény, ezeket az értékeket rendeli a programfüggvény a kezdeti állapotokhoz

A program akkor oldja meg a feladatot, ha minden olyan pontból, amiről a feladat rendelkezik, a megengedett megoldások egyikében állhat csak meg