„Programozási alapismeretek”

Inkább:

Bevezetés a programozáshoz

Alapfogalmak

- Direktszorzat
  - Legyenek $A_1,\dots,A_n$ tetszőleges halmazok. $A = A_1 x \dots x A_n = \{ (a_1,\dots,a_n) \mid a_1 \in A_1,\dots,a_n \in A_n)$
  - Például: $\{1,2,3\} \times \{a,b,c\} = \{(1,a),(1,b),(1,c),(2,a),(2,b),(2,c),(3,a),(3,b),(3,c)\}$
- (Bináris) Reláció
  - $R \subseteq A \times B$
  - Például: $\{(1,b),(2,a),(2,c)\} \subseteq \{1,2,3\} \times \{a,b,c\}$

$(1,b) \in R$

- Logikai reláció
  - $R \subseteq A \times \mathbf{L}$
  - Igazsághalmaz: $[R] = \{ a \in A \mid R(a) = \text{igaz}\}$

Feladat vs. Specifikáció

Feladat

- Természetes nyelven íródott
- Nem egyértelmű
- Nem pontos!
- Ellentmondások lehetnek benne
- **Ebből nem lehet programot készíteni!**

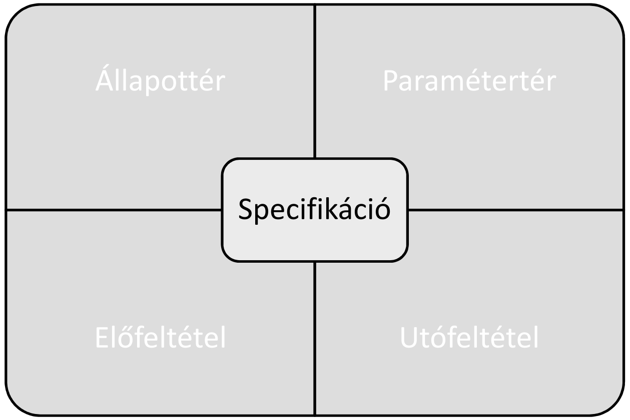
Specifikáció

- Formális vagy fél-formális nyelv
- Egyértelmű
- Ellentmondás mentes
- **Ez alapján már lehet programot készíteni!**

Feladat: Adott két szám, döntsük el, hogy az egyiket elosztva a másikkal 1-et kapunk-e maradékul

Modellezés

- Ha belegondolunk minden alkalommal, amikor programot készítünk, a valós világ dolgait próbáljuk megértetni a számítógéppel
- Azonban a körülöttünk lévő világ végtelenül bonyolult
- Így annak mindig csak egy részletét – a számunkra fontos információt tartalmazó szeletét – vizsgáljuk
- Ezt a folyamatot nevezik modellezésnek, amikor a valós világ (egy részét) megpróbáljuk leírni a rendelkezésre álló eszközökkel
- Erről szól ez a tárgy is: egy (statikus) programozási modell megismeréséről



Állapottér

$$A = \mathbf{N}_m \times \mathbf{N}_n \times \mathbf{L}_l$$

- ??
- Egy direktszorzat
- A fenti példa esetében ilyen elemekből állhat:
  - (1,2,igaz), (22,64,igaz), (1024, 2, hamis), …
- ?
- A feladat megoldásához szükséges változók neve és típusa

Paramétertér

$$B = \mathbf{N}_{m'} \times \mathbf{N}_{n'}$$

- ?
- Szintén egy direktszorzat
- Az állapottér egy altere
- A fenti példa esetében ilyen elemekből állhat:
  - (1,2), (22,64), (1024, 2), …
- Azok a változók, amelyektől függ a végeredmény (Input változók)

Előfeltétel

$$Q = (m=m' \wedge n=n' \wedge n<>0)$$

- ?
- Logikai állítás
- Nullad és elsőrendű logikai kifejezések
- Azt az állapotot írjuk le, amely előzetesen szükséges, hogy a megoldásunk jó legyen
- A fenti példa esetében azt követeljük meg, hogy m és n vegyen fel valamilyen kezdőértéket, és n értéke ne legyen 0.

Utófeltétel

$R = (Q \wedge l = (m \bmod n = 1))$

- Szintén egy logikai állítás
- Nullad és elsőrendű logikai kifejezések
- Azt az állapotot írjuk le, amely a megoldó program lefutása után érvényes. Vagyis a célt amit el kívánunk érni
- A fenti példa esetében azt fogalmazzuk meg, hogy a programunk végén az l változó tartalmazni fogja, hogy ha m-et elosztjuk n-nel, akkor 1-et kapunk-e maradékul (Igaz vagy Hamis)

Feladat vs. Specifikáció

Feladat

Adott két szám, döntsük el, hogy az egyiket elosztva a másikkal 1-et kapunk-e maradékul

Specifikáció

$A = \mathbf{N} \times \mathbf{N} \times \mathbf{L}$

m n l

$B = \mathbf{N} \times \mathbf{N}$

m' n'

$Q = (m=m' \wedge n=n' \wedge n<>0)$

$R = (Q \wedge l = (m \bmod n = 1))$

Észrevételek

- Az állapot –és a paramétertér direktszorzatok
- Az állapottér elemei állapotok:
  - $a \in A$ egy állapot ($A = A1 \times \ldots \times An$)
  - a egy rendezett n-es ($a = (a1,\ldots,an)$), ahol $a_i \in Ai$)
- A változók tekinthetők v: $A \to Ai$ projekciós függvényeknek
- Az elő –és utófeltétel logikai reláció-szerűségek
  - Egész pontosan $A \to \mathbf{L}$ logikai állítások
- Maga a megoldandó feladat tekinthető egy $F \subseteq A \times A$ relációnak, hiszen a feladat előírja, hogy milyen input adatokra milyen outputot kell előállítani. Máshogy fogalmazva: adott állapotból indítva a programot milyen állapot(ok)ba érkezhetünk.

Közeledjünk kicsit az előadáshoz…

- A paramétertér…
  - Nézzük meg, hogy az előző példában az alábbi kezdőállapotokhoz milyen végállapotok tartoznak:
  - (**4,2**,hamis) → (4,2,**hamis**)
(**4,2**,igaz) → (4,2,**hamis**)
(**4,3**,hamis) → (4,3,**igaz**)
(**4,3**,igaz) → (4,3,**igaz**)
  - Vegyük észre, hogy a végállapot nem függ az állapottér összes komponensétől, azaz az állapottér több pontjához is ugyanaz a végállapot tartozik.
  - Formálisan ezeket a pontokat fogjuk össze egyetlen ponttá a paramétertéren

Egy megjegyzés

- A programozási modell megengedi, hogy a feladatok és a programok nem determinisztikusak legyenek!
- A feladatoknál érthető, hiszen sok esetben tényleg nem egy jó megoldás van
- De a programoknál sem követeljük meg azt, hogy ha kétszer elindítjuk ugyanazokkal az értékekkel, akkor pontosan ugyanúgy fusson le!
- Miféle hülyeség ez??
- A valóságban is lehet nem determinisztikus programokat írni!

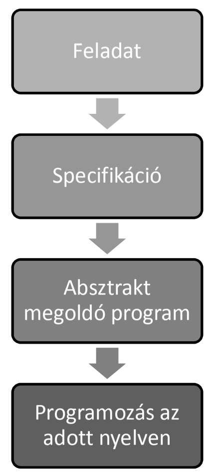
Gondoljuk tovább

- Van tehát egy S programunk, aminek egy előre meghatározott R utófeltételbe kell eljutnia, hiszen ekkor lesz „jó” a program
- Másrészről pedig megadtunk egy Q előfeltételt, de vajon jó az előfeltétel? Biztos igaz az, hogy ha Q teljesül, akkor az S program egy R-nek megfelelő állapotba jut?
- Nem lehetne visszafele gondolkodni?
- Megvan az R utófeltételem és az S programom, a kérdés, hogy ehhez milyen Q előfeltételre van szükségem, hogy „Q-ból indulva S-sel R-be jussak”

Leggyengébb előfeltétel a.k.a lf

- A leggyengébb előfeltétel pontosan azokban az állapotokban igaz, ahonnét indulva S program (biztosan terminál) és az összes lehetséges végállapotára teljesül az R utófeltétel.
- Vagyis megmondja, hogy minek kell ahhoz teljesülnie, hogy S programmal biztos R-be jussak.
- Az lf a legbővebb ilyen halmaz
- Ez a definíció nem zárja ki, hogy az lf által meghatározott halmazon kívüli állapotból indítva a programot az R-beli állapotba jusson!

Hogy lesz ebből az egészből program?



1. feladat

- Keressük meg egy természetes szám egy osztóját!
- A = N x N
  a b
- B = N
  a’
- Q = (a = a’)
- R = (Q $\wedge$ b | a)

2. feladat

- Keressük meg egy természetes szám egy <u>valódi</u> osztóját!
- A = N x N
  a b
- B = N
  a’
- Q = (a = a’)
- R = (Q $\wedge$ b | a $\wedge$ b<>1 $\wedge$ b<>a)
- Megjegyzés: Ez a specifikáció **rossz**, de pár perc múlva felírjuk jól is ☺

3. feladat

- Keressük meg egy <u>összetett</u> természetes szám egy <u>valódi</u> osztóját!
- A = N x N
  a b
- B = N
  a’
- Q = (a = a’ $\wedge$ $\neg$prim(a) $\wedge$ a<>1)
- R = (Q $\wedge$ b | a $\wedge$ b<>1 $\wedge$ b<>a)

4. feladat

- Keressük meg egy ~~összetett~~ természetes szám egy <u>valódi</u> osztóját!
- Megjegyzés: biztos hogy van ilyen?
- A = N x N x L
  a b l
- B = N
  a’
- Q = (a = a’)
- R = (Q $\wedge$ l = (prim(a) $\vee$ a=1) $\wedge$
  $\neg$l $\Rightarrow$ (b | a $\wedge$ b<>1 $\wedge$ b<>a))

5. feladat

- Keressük meg egy természetes szám összes osztóját!
- A = N x H
a h
- B = N
a’
- Q = (a = a’)
- R = (Q $\wedge$ h = { y $\in$ N | y|a })
- Megjegyzés: most lenne gond, ha valódi osztókat keresnénk?

6. feladat

- Keressük meg egy természetes szám legnagyobb prímosztóját
- A = N x N
a m
- B = N
a’
- Q = (a=a’ $\wedge$ a>1)
- R = (Q $\wedge$ m | a $\wedge$ prim(m) $\wedge$
$\forall$ i$\in$ [2..a]: ((prim(i) $\wedge$ i | a) $\Rightarrow$ b>=i))

7. feladat

- Hány valódi osztója van egy adott természetes számnak?
- A = N x N
a d
- B = N
a’
- Q = (a=a’)
- R = Q $\wedge$ d = $\sum \mathcal{X}$(i|a)

Köszönöm a figyelmet!