

Bevezetés az informatikába

6. előadás

Dr. Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Matematikus BSc - I. félév / 2008 / Budapest



1 Alapfogalmak

- típus
- állapottér, feladat
- program, megoldás

2 Specifikáció

- állapottér, előfeltétel, utófeltétel

3 Elemi programok és szerkezetek

- strukturált programok
- elemi programok
- programszerkezetek

Tartalom

1 Alapfogalmak

- típus
- állapottér, feladat
- program, megoldás

2 Specifikáció

- állapottér, előfeltétel, utófeltétel

3 Elemi programok és szerkezetek

- strukturált programok
- elemi programok
- programszerkezetek

Többszintű számítógépek (Andrew S. Tannenbaum)

- 5. problémaorientált nyelvi szint (magas szintű nyelv)
- 4. assembly nyelvi szint
- 3. operációs rendszer gépi szintje
- 2. utasításrendszer-architektúra szintje
- 1. mikroarchitektúra szintje
- 0. digitális logikai szint

*"Hogyan tudunk
problémaorientált (magas) nyelvi szintű
programot készíteni?"*

Programozás, programtervezés

Programozás

"A programozás az a tevékenység, amikor bizonyos **feladatok megoldására programokat** készítünk,...
amelyeket azután **számítógépekkel** hajtunk végre."

programtervezés \neq **kódolás**

Programtervezés

Programtervezés :

- megoldandó problémára összpontosít
- független a programozási környezettől
- az alkalmazott módszerek matematikai alapossággal igazolhatók

Tartalom

- 1 **Alapfogalmak**
 - **típus**
 - állapotter, feladat
 - program, megoldás
- 2 **Specifikáció**
 - állapotter, előfeltétel, utófeltétel
- 3 **Elemi programok és szerkezetek**
 - strukturált programok
 - elemi programok
 - programszerkezetek

Típus

típus = felvehető értékek halmaza + elvégezhető műveletek

Adat típusok :

típus	jelölés	értékhalmoz	műveletek
természetes szám	\mathbb{N}	pozitív egész számok és a 0	$+, -, *, /, =, >, <, \dots$
egész szám	\mathbb{Z}		$+, -, *, /, =, >, <, \dots$
valós szám	\mathbb{R}		$+, -, *, /, =, >, <, \dots$
logikai	\mathbb{L}	igaz, hamis	\wedge, \vee, \neg
karakter		a,b,c,... A,B,C,...	$>, <, =, \dots$
halmoz	2^E	E alaphalmaz	$\cap, \cup, \text{kivétel, hozzáadás, } \dots$
sorozat	E^*	E alaphalmaz	hossz, hivatkozás, törlés, ...
vektor			
mátrix			

összetett típusok...

Tartalom

1 Alapfogalmak

- típus
- **állapottér, feladat**
- program, megoldás

2 Specifikáció

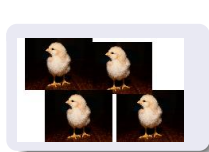
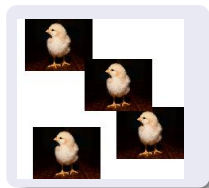
- állapottér, előfeltétel, utófeltétel

3 Elemi programok és szerkezetek

- strukturált programok
- elemi programok
- programszerkezetek

Állapottér

- feladat egy állapota: a feladat adatainak az értékei
- **állapottér** : az összes lehetséges állapot



- a **feladat** egy leképzés : a kezdőállapot(ok)ból a célállapot(ok)ba

Tartalom

1 Alapfogalmak

- típus
- állapottér, feladat
- **program, megoldás**

2 Specifikáció

- állapottér, előfeltétel, utófeltétel

3 Elemi programok és szerkezetek

- strukturált programok
- elemi programok
- programszerkezetek

Program

Mi a program ?

"A program utasítások sorozata" ?

utasítás készlet ?

programozási nyelv ?

utasításokat értelmező gép ?

Program

Mi a program ?

"A program utasítások sorozata" ?

utasítás készlet ?

programozási nyelv ?

utasításokat értelmező gép ?

Program

Mi a program ?

"A program utasítások sorozata" ?

utasítás készlet ?

programozási nyelv ?

utasításokat értelmező gép ?

Program

Mi a program ?

"A program utasítások sorozata" ?

utasítás készlet ?

programozási nyelv ?

utasításokat értelmező gép ?

Program

Mi a program ?

"A program utasítások sorozata" ?

utasítás készlet ?

programozási nyelv ?

utasításokat értelmező gép ?

Program

Állapottér $n:\mathbb{N}$, $d:\mathbb{N}$, program utasítások :

1. Legyen d értéke vagy $n-1$ vagy 2 .
2. Ha d értéke megegyezik 2 -vel
akkor: Amíg d nem osztja $n-t$
addig növeljük d értékét 1 -gyel,
különben: Amíg d nem osztja $n-t$
addig csökkentjük d értékét 1 -gyel.

- kiindulási állapot $(n:6,d:8)$
- első utasítás $(n:6,d:2)$ vagy $(n:6,d:5)$
- második utasítás vagy $(n:6,d:2)$ és leáll, célállapot
vagy $(n:6,d:4)$ következő $(n:6,d:3)$ és leáll, célállapot

$(6,8)$:

- $\langle(6,8),(6,2)\rangle$
- $\langle(6,8),(6,5) (6,4) (6,3)\rangle$

Program

1. Legyen d értéke vagy $n-1$ vagy 2 .
2. Ha d értéke megegyezik 2 -vel
akkor: Amíg d nem osztja n -t
addig növeljük d értékét 1 -gyel,
különben: Amíg d nem osztja n -t
addig csökkentjük d értékét 1 -gyel.

(5,y) :

- $\langle (5,y) (5,2) (5,3) (5,4) (5,5) \rangle$
- $\langle (5,y) (5,4) (5,3) (5,2) (5,1) \rangle$

(1,y) :

- $\langle (1,y) (1,0) (1,0) \dots \rangle$
- $\langle (1,y) (1,2) (1,3) (1,4) \dots \rangle$

(4,y) :

- $\langle (4,y) (4,2) \rangle$
- $\langle (4,y) (4,3) (4,2) \rangle$

(0,y) :

- $\langle (0,y) (0,y) (0,y) \dots \rangle$
- $\langle (0,2) \rangle$

Program

Program

- A **program** az általa befutható összes lehetséges végrehajtást jelenti.
- Állapotterének bármelyik állapotból el tud indulni, és ahhoz olyan végrehajtási sorozatot rendel, amelyik első állapota a kiinduló állapot.
- Egy kiinduló állapothoz akár több végrehajtási sorozat is tartozhat. Az, hogy ilyenkor éppen melyik végrehajtás következik be, nem definiált, azaz a program nem-determinisztikus.

(15,y) :

- $\langle (15,2) (15,3) \rangle$
- $\langle (15,14) (15,13) (15,12) (15,11) (15,10) (15,9) (15,8) (15,7) (15,6) (15,5) \rangle$

Megoldás

Egy program megold egy feladatot, ha:

- a feladat bármely kezdőállapotból terminál,
- olyan célállapotban áll meg, amelyet a feladat az adott kezdőállapothoz rendel

Tartalom

- 1 Alapfogalmak
 - típus
 - állapottér, feladat
 - program, megoldás
- 2 Specifikáció
 - állapottér, előfeltétel, utófeltétel
- 3 Elemi programok és szerkezetek
 - strukturált programok
 - elemi programok
 - programszerkezetek

Tartalom

- 1 Alapfogalmak
 - típus
 - állapottér, feladat
 - program, megoldás
- 2 Specifikáció
 - állapottér, előfeltétel, utófeltétel
- 3 Elemi programok és szerkezetek
 - strukturált programok
 - elemi programok
 - programszerkezetek

Feladat specifikációja

Feladat specifikációja

- állapotér, A , adatokhoz tartozó változó nevek
- **előfeltétel**, Q , bemenő adatértékek ($'$), kezdőállapotot leíró logikai állítás
- **utófeltétel**, R , kezdőértékhez rendelt célállapotot leíró logikai állítás

Példa

Feladat: egyenes vonalú egyenletes mozgást végző testnek a megtett út (s') és az eltelt idő (t') függvényében kell az átlagsebességét (v) kiszámolni.

$$A = (s : \mathbb{R}, t : \mathbb{R}, v : \mathbb{R})$$

$$Q = (s = s' \text{ és } t = t' \text{ és } s' \geq 0 \text{ és } t' \geq 0)$$

$$R = (v = s'/t')$$

Feladat : egy összetett szám legnagyobb valódi osztóját keressük.

Az adott természetes összetett szám (n'), a keresett valódi osztó (d)
Mindkét adat természetes szám típusú.

Az **állapottér** : $A = (n : \mathbb{N}, d : \mathbb{N})$

A két adat közül az első a bemeneti, a második a kimeneti adat.

Legyen n' a tetszőlegesen kiválasztott bemenő érték.

Mivel a feladat nem értelmes a 0-ra, az 1-re és a prímszámokra; ezért az n' csak összetett szám lehet.

Az **előfeltétel** : $Q = (n = n' \text{ és } n' \text{ összetett szám})$

A célállapot első komponensének, azaz a bemeneti adatnak értéke ne változzon meg, maradjon továbbra is n' ;

a második adat pedig legyen az n' legnagyobb valódi osztója:

Az **utófeltétel** : $R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$

Feladat : egy összetett szám legnagyobb valódi osztóját keressük.
Az adott természetes összetett szám (n'), a keresett valódi osztó (d)
Mindkét adat természetes szám típusú.

Az **állapottér** : $A = (n : \mathbb{N}, d : \mathbb{N})$

A két adat közül az első a bemeneti, a második a kimeneti adat.

Legyen n' a tetszőlegesen kiválasztott bemenő érték.

Mivel a feladat nem értelmes a 0-ra, az 1-re és a prímszámokra; ezért az n' csak összetett szám lehet.

Az **előfeltétel** : $Q = (n = n' \text{ és } n' \text{ összetett szám})$

A célállapot első komponensének, azaz a bemeneti adatnak értéke ne változzon meg, maradjon továbbra is n' ;

a második adat pedig legyen az n' legnagyobb valódi osztója:

Az **utófeltétel** : $R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$

Feladat : egy összetett szám legnagyobb valódi osztóját keressük.
Az adott természetes összetett szám (n'), a keresett valódi osztó (d)
Mindkét adat természetes szám típusú.

Az **állapotér** : $A = (n : \mathbb{N}, d : \mathbb{N})$

A két adat közül az első a bemeneti, a második a kimeneti adat.

Legyen n' a tetszőlegesen kiválasztott bemenő érték.

Mivel a feladat nem értelmes a 0-ra, az 1-re és a prímszámokra; ezért az n' csak összetett szám lehet.

Az **előfeltétel** : $Q = (n = n' \text{ és } n' \text{ összetett szám})$

A célállapot első komponensének, azaz a bemeneti adatnak értéke ne változzon meg, maradjon továbbra is n' ;

a második adat pedig legyen az n' legnagyobb valódi osztója:

Az **utófeltétel** : $R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$

Feladat : egy összetett szám legnagyobb valódi osztóját keressük.
Az adott természetes összetett szám (n'), a keresett valódi osztó (d)
Mindkét adat természetes szám típusú.

Az **állapotér** : $A = (n : \mathbb{N}, d : \mathbb{N})$

A két adat közül az első a bemeneti, a második a kimeneti adat.

Legyen n' a tetszőlegesen kiválasztott bemenő érték.

Mivel a feladat nem értelmes a 0-ra, az 1-re és a prímszámokra; ezért az n' csak összetett szám lehet.

Az **előfeltétel** : $Q = (n = n' \text{ és } n' \text{ összetett szám})$

A célállapot első komponensének, azaz a bemeneti adatnak értéke ne változzon meg, maradjon továbbra is n' ;

a második adat pedig legyen az n' legnagyobb valódi osztója:

Az **utófeltétel** : $R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$

Feladat : egy összetett szám legnagyobb valódi osztóját keressük.
Az adott természetes összetett szám (n'), a keresett valódi osztó (d)
Mindkét adat természetes szám típusú.

Az **állapotér** : $A = (n : \mathbb{N}, d : \mathbb{N})$

A két adat közül az első a bemeneti, a második a kimeneti adat.
Legyen n' a tetszőlegesen kiválasztott bemenő érték.

Mivel a feladat nem értelmes a 0-ra, az 1-re és a prímszámokra; ezért az n' csak összetett szám lehet.

Az **előfeltétel** : $Q = (n = n' \text{ és } n' \text{ összetett szám})$

A célállapot első komponensének, azaz a bemeneti adatnak értéke ne változzon meg, maradjon továbbra is n' ;

a második adat pedig legyen az n' legnagyobb valódi osztója:

Az **utófeltétel** : $R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$

Feladat : egy összetett szám legnagyobb valódi osztóját keressük.
Az adott természetes összetett szám (n'), a keresett valódi osztó (d)
Mindkét adat természetes szám típusú.

Az **állapotér** : $A = (n : \mathbb{N}, d : \mathbb{N})$

A két adat közül az első a bemeneti, a második a kimeneti adat.

Legyen n' a tetszőlegesen kiválasztott bemenő érték.

Mivel a feladat nem értelmes a 0-ra, az 1-re és a prímszámokra; ezért az n' csak összetett szám lehet.

Az **előfeltétel** : $Q = (n = n' \text{ és } n' \text{ összetett szám})$

A célállapot első komponensének, azaz a bemeneti adatnak értéke ne változzon meg, maradjon továbbra is n' ;

a második adat pedig legyen az n' legnagyobb valódi osztója:

Az **utófeltétel** : $R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$

Feladat : egy összetett szám legnagyobb valódi osztóját keressük.
Az adott természetes összetett szám (n'), a keresett valódi osztó (d)
Mindkét adat természetes szám típusú.

Az **állapotér** : $A = (n : \mathbb{N}, d : \mathbb{N})$

A két adat közül az első a bemeneti, a második a kimeneti adat.

Legyen n' a tetszőlegesen kiválasztott bemenő érték.

Mivel a feladat nem értelmes a 0-ra, az 1-re és a prímszámokra; ezért az n' csak összetett szám lehet.

Az **előfeltétel** : $Q = (n = n' \text{ és } n' \text{ összetett szám })$

A célállapot első komponensének, azaz a bemeneti adatnak értéke ne változzon meg, maradjon továbbra is n' ;

a második adat pedig legyen az n' legnagyobb valódi osztója:

Az **utófeltétel** : $R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek })$

Specifikáció - összetett szám legnagyobb osztója

$$A = (n : \mathbb{N}, d : \mathbb{N})$$

$$Q = (n = n' \text{ és } n' \text{ összetett szám})$$

$$R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$$

$$A = (n : \mathbb{N}, d : \mathbb{N})$$

$$Q = (n = n' \wedge \exists k \in [2..n-1] : k|n)$$

$$R = (Q \wedge d \in [2..n-1] \wedge d|n \wedge \forall k \in [d+1..n-1] : \neg k|n)$$

$$[m..n] = [m, n] \cap \mathbb{Z}$$

Specifikáció - összetett szám legnagyobb osztója

$$A = (n : \mathbb{N}, d : \mathbb{N})$$

$$Q = (n = n' \text{ és } n' \text{ összetett szám})$$

$R = (Q \text{ és } d \text{ valódi osztója } n\text{-nek és bármelyik olyan } k \text{ szám esetén, amelyik nagyobb mint a } d, \text{ a } k \text{ nem valódi osztója } n\text{-nek})$

$$A = (n : \mathbb{N}, d : \mathbb{N})$$

$$Q = (n = n' \wedge \exists k \in [2..n-1] : k|n)$$

$$R = (Q \wedge d \in [2..n-1] \wedge d|n \wedge \forall k \in [d+1..n-1] : \neg k|n)$$

$$[m..n] = [m, n] \cap \mathbb{Z}$$

Tartalom

- 1 Alapfogalmak
 - típus
 - állapottér, feladat
 - program, megoldás
- 2 Specifikáció
 - állapottér, előfeltétel, utófeltétel
- 3 Elemi programok és szerkezetek
 - strukturált programok
 - elemi programok
 - programszerkezetek

Tartalom

- 1 Alapfogalmak
 - típus
 - állapottér, feladat
 - program, megoldás
- 2 Specifikáció
 - állapottér, előfeltétel, utófeltétel
- 3 Elemi programok és szerkezetek
 - **strukturált programok**
 - elemi programok
 - programszerkezetek

Strukturált programok

Elemi programok

- 1 üres program
- 2 rossz program
- 3 értékadás

Program szerkezetek

- 1 szekvencia
- 2 elágazás
- 3 ciklus

Tartalom

- 1 Alapfogalmak
 - típus
 - állapottér, feladat
 - program, megoldás
- 2 Specifikáció
 - állapottér, előfeltétel, utófeltétel
- 3 Elemi programok és szerkezetek
 - strukturált programok
 - **elemi programok**
 - programszerkezetek

Üres program - SKIP

SKIP

Bármelyik állapotból indul el, abban az állapotban marad

$$A = (a : \mathbb{N}, b : \mathbb{N}, c : \mathbb{N})$$

$$Q = (a = 2 \wedge b = 5 \wedge c = 7)$$

$$R = (Q \wedge (c = a + b \vee c = 10))$$

Rossz program - ABORT

ABORT

Bármelyik állapotból indul el, abnormális végrehajtást kezd el

Értékadás

$$x, y := y, x$$

Egy vagy több változó új értéket kap.

$$A = (x : \mathbb{Z}, y : \mathbb{Z})$$

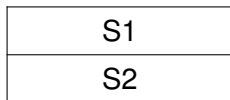
$$Q = (x = x' \wedge y = y')$$

$$R = (x = y' \wedge y = x')$$

Tartalom

- 1 Alapfogalmak
 - típus
 - állapottér, feladat
 - program, megoldás
- 2 Specifikáció
 - állapottér, előfeltétel, utófeltétel
- 3 Elemi programok és szerkezetek
 - strukturált programok
 - elemi programok
 - **programszerkezetek**

Szekvencia

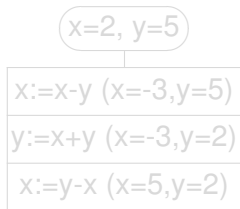
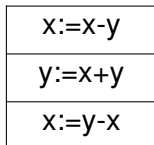


Az egyik program végrehajtása után közvetlenül a másikat hajtódik végre.

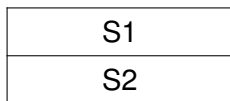
$$A = (x : \mathbb{Z}, y : \mathbb{Z})$$

$$Q = (x = x' \wedge y = y')$$

$$R = (x = y' \wedge y = x')$$



Szekvencia

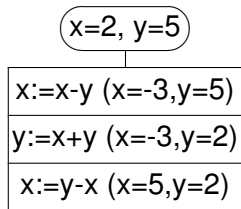
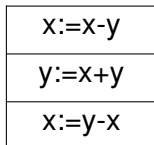


Az egyik program végrehajtása után közvetlenül a másikat hajtódik végre.

$$A = (x : \mathbb{Z}, y : \mathbb{Z})$$

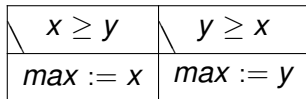
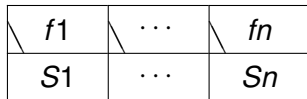
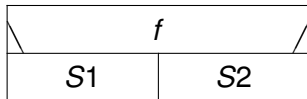
$$Q = (x = x' \wedge y = y')$$

$$R = (x = y' \wedge y = x')$$



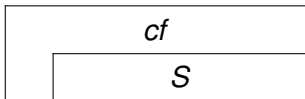
Elágazás

Két vagy több programhoz egy-egy feltételt rendelünk. Az a program (programág) hajtódik végre, amelyhez tartozó feltétel az aktuális állapotban teljesül.



Ciklus

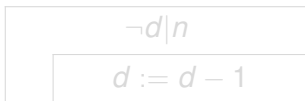
Egy adott program (ciklusmag) egymás után annyszor hajtódik végre, valahányszor olyan állapotban állunk, amelyre egy adott feltétel (ciklusfeltétel) teljesül.



$$A = (n : \mathbb{N}, d : \mathbb{R})$$

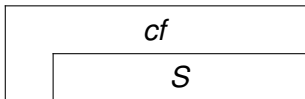
$$Q = (n = n' \wedge \exists k \in [2..n-1] : k|n \wedge d = n-1)$$

$$R = (n = n' \wedge d \in [2..n-1] \wedge d|n \wedge \forall k \in [d+1..n-1] : \neg k|n)$$



Ciklus

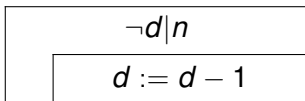
Egy adott program (ciklusmag) egymás után annyszor hajtódik végre, valahányszor olyan állapotban állunk, amelyre egy adott feltétel (ciklusfeltétel) teljesül.



$$A = (n : \mathbb{N}, d : \mathbb{R})$$

$$Q = (n = n' \wedge \exists k \in [2..n-1] : k|n \wedge d = n-1)$$

$$R = (n = n' \wedge d \in [2..n-1] \wedge d|n \wedge \forall k \in [d+1..n-1] : \neg k|n)$$



Összefoglalás

"Programkészítés":

- 1 szöveges feladat megadás
- 2 specifikáció
- 3 struktogramm (elemi programok + programszerkezetek)
- 4 magas szintű nyelvi kód (forrás)
- 5 "futtatható kód"

programozási tételek...

Ajánlott irodalom :

- Bevezetés a programozásba (Fóthi Ákos)

<http://people.inf.elte.hu/ekonyvtar/>

- Programozás módszertan (Gregorics Tibor)

<http://people.inf.elte.hu/gt/prog/konyv/konyv.html>