



ELTE | IK

PROGRAMOZÁS

10. előadás

Horváth Győző, Horváth Gyula, Szlávi Péter



Állapottér-specifikáció kitekintés



Probléma

- Hogyan tudjuk jobban leírni, ha egy adatból nem előáll egy másik, hanem megváltozik?

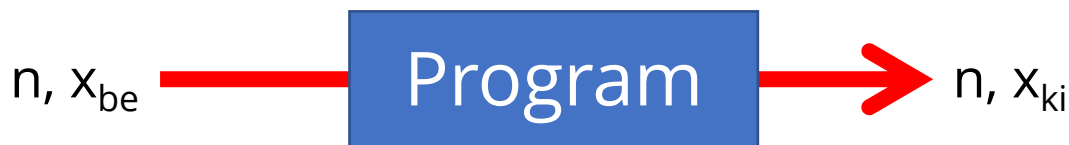
- Eddig:



- Például:



- De hogyan? (pl. helyben rendezés)



Példa

Cseréljük fel két változó értékét!

Specifikáció:

Be: $abe \in \mathbb{Z}$, $bbe \in \mathbb{Z}$

Ki: $aki \in \mathbb{Z}$, $bki \in \mathbb{Z}$

Ef: -

Uf: $aki = bbe$ és $bki = abe$

Például:

Kezdetben: $(a:5, b:3)$

Végén: $(a:3, b:5)$

Példa

Cseréljük fel két változó értékét!

Specifikáció:

Be: $a \in \mathbb{Z}, b \in \mathbb{Z}$

Ki: $a' \in \mathbb{Z}, b' \in \mathbb{Z}$

Ef: -

Uf: $a' = b$ és $b' = a$

Például:

Kezdetben: $(a:5, b:3)$

Végén: $(a:3, b:5)$

Adatspecifikáció

- **Adat** mint egy értékhalmoz eleme
(+ értékhalmozhoz „szokásosan” asszociált műveletek)
 - Pl. $a \in \mathbb{Z}$ (aritmetikai műveletekkel)
- Bemenő adatok, kimenő adatok
- **Feladat**: olyan előírás (reláció), amely megadja, hogy adott bemenetre adott kimenetet mikor fogadjuk el helyes kimenetként (tehát a feladat megoldásaként)
 - Bemenő adatok \rightarrow Kimenő adatok

Állapottér-specifikáció

- **Adat típusa:** értékhalmoz + műveletek
 - Pl. a : Egész vagy $a:Z$
- Egy **állapot**: feladat minden adata felvesz egy-egy értéket
 - Pl. (a : 5, b : 3)
- **Állapottér:** összes lehetséges állapot
 - Változók: az állapottér adatainak „címkéi”
 - Pl. (a : Egész, b : Egész) vagy
 - Pl. ($a:Z$, $b:Z$)

Állapottér-specifikáció

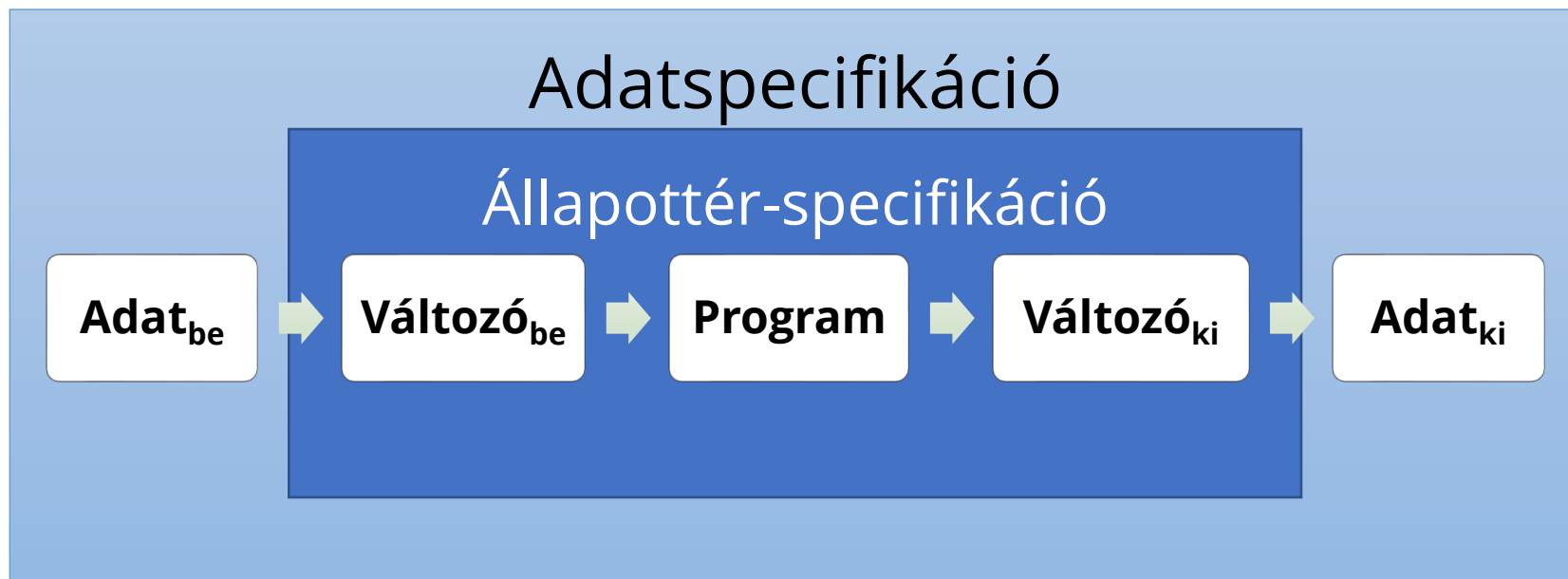
- Kezdőállapot, célállapot
- **Feladat:** olyan előírás (reláció), amely megadja, hogy adott kezdőállapotra adott végállapotot mikor fogadjuk el helyesnek (tehát a feladat megoldásaként)
- **Végrehajtás:** az állapottér változóinak módosítása
 - Pl: $(a:3, b:5) \rightarrow (a:4, b:5)$
- **Program:** egy végrehajtási sorozat, mely a kezdőállapotból indul és a célállapotban ér véget.
 - Pl: $(a:3, b:5) \rightarrow (a:3, b:5, sv:3) \rightarrow (a:5, b:5, sv:3) \rightarrow (a:5, b:3, sv:3) \rightarrow (a:5, b:3)$

sv : segédváltozó
állapottér-bővítés

Állapottér-specifikáció

- Állapottér
 - A feladat lényeges adatainak típusérték-halmazai az egyes adatokhoz tartozó változónevekkel együtt
 - Bemeneti és kimeneti változók
- Előfeltétel
 - kezdőállapotok halmazát leíró logikai állítás
 - rögzíti a bemenő változók egy lehetséges, de tetszőleges kezdőértékét
- Utófeltétel
 - Logikai állítás, amely megadja, hogy adott kezdőállapothoz milyen végállapot lehet helyes megoldás

Állapottér-specifikáció sematikus ábra



Állapottér-specifikáció

Példa

Cseréljük fel két változó értékét!

Adatspecifikáció:

Be: $a \in \mathbb{Z}, b \in \mathbb{Z}$

Ki: $a' \in \mathbb{Z}, b' \in \mathbb{Z}$

Ef: -

Uf: $a' = b$ és $b' = a$



Állapottér-specifikáció:

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

$Ef = (a = a' \text{ és } b = b')$

$Uf = (a = b' \text{ és } b = a')$

Például:

Kezdetben: $(a:5, b:3)$

Végén: $(a:3, b:5)$

- Állapottér
 - A feladat lényeges adatainak típusérték-halmazai az egyes adatokhoz tartozó változónevekkel együtt
 - Bemeneti és kimeneti változók
- Előfeltétel
 - kezdőállapotok halmazát leíró logikai állítás
 - rögzíti a bemenő változók egy lehetséges, de tetszőleges kezdőértékét
- Utófeltétel
 - Logikai állítás, amely megadja, hogy adott kezdőállapothoz milyen végállapot lehet helyes megoldás

Más a' jelentése!

Adatspecifikációban: a kimeneti adatot jelöli

Állapottér-specifikációban: a változók kezdőértékét jelöli.

Lehetne a_0 és b_0 is, de a megadott irodalom így jelöli.

Állapottér-specifikáció

Példa

Cseréljük fel két változó értékét!

Állapottér-specifikáció:

$A = (a:Z, b:Z)$

$A = (a:\text{Egész}, b:\text{Egész})$

$Ef = (a=a' \text{ és } b=b')$

$Uf = (a=b' \text{ és } b=a')$

Algoritmus:

<code>sv:=a</code>
<code>a:=b</code>
<code>b:=sv</code>

Változó
`sv:Egész`

Állapottér-specifikáció

Példa

Növeljük meg egy változó értékét!

Állapottér-specifikáció:

$A = (a : \mathbb{Z})$

$E_f = (a = a')$

$U_f = (a = a' + 1)$

Algoritmus:

```
a:=a+1
```

Állapottér-specifikáció

Példa

Alakítsunk át egy Celsius értéket Fahrenheitté!

Állapottér-specifikáció:

$A = (c : \mathbb{R}, f : \mathbb{R})$

$Ef = (c = c')$

$Uf = (c = c' \text{ és } f = c * 9 / 5 + 32)$

$Uf = (Ef \text{ és } f = c * 9 / 5 + 32)$

A bemeneti változó (c) értéke (c') nem változik a megoldás során, a program végén is ugyanaz az értéke, mint kezdetben (c=c').

Állapottér-specifikáció

Eldöntés

Állapottér-specifikáció:

$A = (e:Z, u:Z, \text{van}:L)$

$Ef = (e=e' \text{ és } u=u')$

$Uf = (Ef \text{ és } \text{van} = \exists i \in [e..u] : (T(i)))$

$Uf = (Ef \text{ és } \text{van} = \text{VAN}(i=e..u, T(i)))$

A bemeneti változók értékei nem változnak a megoldás során, a program végén is ugyanaz az értékük, mint kezdetben.

Specifikáció

$Be: e \in Z, u \in Z$

$Ki: \text{van} \in L$

$Ef: -$

$Uf: \text{van} = \exists i \in [e..u] : (T(i))$

Rövidítve:

$Uf: \text{van} = \text{VAN}(i=e..u, T(i))$

Összehasonlítás

Adatspecifikáció	Állapottér-specifikáció	Rövidítés
Adat	Változó	
Halmaz	Típus	
$a \in Z$	$a:\text{Egész}$	$a:Z$
Külön: Be, Ki	Egyben: A (az állapottér)	
Be: $a \in Z$ Ki: $b \in Z$	$A=(a:\text{Egész}, b:\text{Egész})$	$A=(a:Z, b:Z)$
Sorozat: $a \in Z[1..5]$	Tömb: $a:\text{Tömb}[1..5:\text{Egész}]$	$a:Z[1..5]$
Direkt szorzat: $a \in \text{Pont}$, $\text{Pont} = X \times Y$, $X = N$, $Y = N$	Rekord: $a:\text{Pont}$, $\text{Pont} = \text{Rekord}(x:\text{Egész}, y:\text{Egész})$	
Ef: -	$Ef=(a=a')$	
Uf: ... (ha a bemenet nem változik)	$Uf=(Ef \text{ és } \dots)$ (ha a bemenet nem változik)	

Állapottér-specifikáció

Példa

Egy tanuló év végi osztályzatai alapján állapítsuk meg, hogy bukott-e!

Állapottér-specifikáció:

$A = (n:N, \text{jegyek}:N[1..n], \text{bukott}:L)$

$Ef = (n=n' \text{ és } \text{jegyek}=\text{jegyek}' \text{ és } \forall i \in [1..n]: (1 \leq \text{jegyek}[i] \leq 5))$

$Uf = (Ef \text{ és } \text{bukott}=\text{VAN}(i=1..n, \text{jegyek}[i]=1))$

A bemeneti változók értékei nem változnak a megoldás során, a program végén is ugyanaz az értékük, mint kezdetben.

Bukott-e

$Be: n \in N, \text{jegyek} \in N[1..n]$

$Ki: \text{bukott} \in L$

$Ef: \forall i \in [1..n]: (1 \leq \text{jegyek}[i] \leq 5)$

$Uf: \text{bukott}=\text{VAN}(i=1..n, \text{jegyek}[i]=1)$

Eldöntés:

$A = (e:Z, u:Z, \text{van}:L)$

$Ef = (e=e' \text{ és } u=u')$

$Uf = (Ef \text{ és } \text{van}=\text{VAN}(i=e..u, T(i)))$

van	~	bukott
e..u	~	1..n
T(i)	~	jegyek[i]=1

Akasztófa játék

- Állapottér (amivel a játék leírható, működtethető)
 - kitalálandó szó
 - korábbi tippelt betűk
- Műveletek
 - tippelés
- Lekérdezések (rajzolás, ellenőrzés)
 - a szó egy karakterére tippeltünk-e már
 - rossz tippek száma
 - nyertünk-e
 - veszítettünk-e



Akasztófa játék

- Állapottér (amivel a játék leírható, működtethető)
 - kitalálendő szó
 - korábbi tippelt betűk tömbje

$A = (\text{szó}:S, n:N, \text{tippek}:K[1..n])$



```
static void Main(string[] args) {  
    string szo = "alma";  
    List<char> tippek = new List<char>() { 'a', 'b', 'c' };  
}
```

Akasztófa játék

- Típpelés

- a tippelt karakterre nem tippelhettünk korábban
- a tippelt karakter a tömb végére kerül

Specifikáció

$A = (\text{szó}:S, n:N, \text{tippek}:K[1..n], \text{tipp}:K)$

$Ef = (\text{szó}=\text{szó}' \text{ és } n=n' \text{ és } \text{tippek}=\text{tippek}' \text{ és } \text{tipp}=\text{tipp}' \text{ és } \forall i \in [1..n]: (\text{tipp} \neq \text{tippek}[i]))$

$Uf = (\text{szó}=\text{szó}' \text{ és } \text{tipp}=\text{tipp}' \text{ és } \forall i \in [1..n']: (\text{tippek}[i] = \text{tippek}'[i]) \\ n=n'+1 \text{ és } \text{tippek}[n]=\text{tipp})$

```
static void tippelés(string szo, List<char> tippek,
                    char tipp) {
    tippek.Add(tipp);
}
```

```
a b l a k _ _ e _
Hibak szama: 6/9

+---+
|   |
| 0  |
|   |
|   |
+---+

a e s t k r b n l z p
Tipp: |
```

Akasztófa játék

- Típpeltünk-e már a szó i . betűjére?
 - benne van-e a szó i . betűje a tippek tömbben?
 - eldöntés

Specifikáció

$A = (\text{szó}:S, n:N, \text{tippek}:K[1..n], i:N, \text{van}:L)$

$Ef = (\text{szó}=\text{szó}' \text{ és } n=n' \text{ és } \text{tippek}=\text{tippek}' \text{ és } i=i')$

$Uf = (Ef \text{ és } \text{van}=\text{VAN}(j=1..n, \text{szó}[i]=\text{tippek}[j]))$

```
static bool tippelte(string szo, List<char> tippek, int i) {  
    return Mintak.Van(1, tippek.Count,  
                      j => szo[i - 1] == tippek[j - 1]);  
}
```

a b l a k _ _ e _
Hibak szama: 6/9

```
+---+  
|   |  
| 0  |  
|   |  
|   |  
+---+
```

a e s t k r b n l z p
Tipp: |

Akasztófa játék

- Rossz tippek száma
 - **hány** olyan tipp van, ami nincs **benne** a szóban?
 - **megszámolás**ban **eldöntés**

Specifikáció

A =(szó:S, n:N, tippek:K[1..n], **db:N**)

Fv=(szóban:K->L, szóban(b)=**VAN**(i=1..hossz(szó), b=szó[i]))

Ef=(szó=szó' és n=n' és tippek=tippek')

Uf=(**Ef és** db=**DARAB**(i=1..n, nem szóban(tippekek[i]))))

```
static int rosszTippekSzama(string szo, List<char> tippek) {  
    return Mintak.Darab(1, tippek.Count,  
        i => !Mintak.Van(1, szo.Length,  
            j => tippek[i - 1] == szo[j - 1]));  
}
```

a b l a k _ _ e _
Hibák száma: 6/9

```
+---+  
|   |  
| 0  |  
|   |  
/    |  
-----
```

a e s t k r b n l z p
Tipp: |

Akasztófa játék

- Nyertünk-e?
 - a szó **minden** betűje **benne** van-e a tippek tömbben?
 - **mind eldöntés**ben **eldöntés**

Specifikáció

A =(szó:S, n:N, tippek:K[1..n], **nyert:L**)

Fv=(benne:K->L, benne(b)=VAN(i=1..n, b=tippek[i]))

Ef=(szó=szó' és n=n' és tippek=tippek')

Uf=(**Ef és** nyert=**MIND**(i=1..hossz(szó), benne(szó[i]))))

```
static bool nyertE(string szo, List<char> tippek) {  
    return Mintak.Mind(1, szo.Length,  
        i => Mintak.Van(1, tippek.Count,  
            j => szo[i - 1] == tippek[j - 1]));  
}
```

a b l a k _ _ e _
Hibák száma: 6/9

```
+---+  
|   |  
| 0 |  
|   |  
/   |  
+---+
```

a e s t k r b n l z p
Tipp: |