

PROGRAMOZÁS Állapottér-specifikáció

Horváth Győző



Állapottér-specifikáció kitekintés



Probléma

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

Eddig:

 Bemeneti adatok
 Program
 Kimeneti adatok

 Például:

 e, u
 Program
 van, ind

De hogyan? (pl. helyben rendezés)

n,
$$x_{be}$$
 Program \rightarrow n, x_{ki}



Példa

Cseréljük fel két változó értékét!

Specifikáció:

```
Be: abe∈Z, bbe∈Z
Ki: aki∈Z, bki∈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∈Z, b∈Z
Ki: a'∈Z, b'∈Z
Ef: -
Uf: a'=b és b'=a
```

Például:

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

Adatspecifikáció

- Adat mint egy értékhalmaz eleme (+ értékhalmazhoz "szokásosan" asszociált műveletek)
 - Pl. a∈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 → Kimenő adatok

- Adat típusa: értékhalmaz + 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)

- 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) → (a:4, b:5)

sv : segédváltozó állapottér-bővítés

- 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)

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



Cseréljük fel két változó értékét!

- 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

Adatspecifikáció:

Be: $a \in Z$, $b \in Z$

Ki: $a' \in Z$, $b' \in Z$

Ef: -

Uf: a'=b és b'=a

Állapottér-specifikáció:

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

Például:

Kezdetben: (a:5, b:3)

Végén: (a:3, b:5)

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.



Cseréljük fel két változó értékét!

Állapottér-specifikáció:

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

Ef=(a=a' és b=b')

Uf=(a=b' és b=a')
```

Algoritmus:

```
sv:=a

a:=b

b:=sv
```

Növeljük meg egy változó értékét!

Állapottér-specifikáció:

```
A =(a:Z)
Ef=(a=a')
Uf=(a=a'+1)
```

Algoritmus:

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

Alakítsunk át egy Celsius értéket Fahrenheitté!

Állapottér-specifikáció:

```
A =(c:R, f:R)

Ef=(c=c')

Uf=(c=c' és f=c*9/5+32)

Uf=(Ef 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, van:L)

Ef=(e=e' és u=u')

Uf=(Ef és van=∃i∈[e..u]:(T(i)))

Uf=(Ef és van=VAN(i=e..u, T(i)))
```

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

```
Be: e∈Z, u∈Z
    Ki: van∈L
    Ef: -
    Uf: van=∃i∈[e..u]:(T(i))
    Rövidítve:
    Uf: van=VAN(i=e..u,T(i))
))
))
```

Összehasonlítás

Adatspecifikáció	Állapottér-specifikáció	Rövidítés
Adat	Változó	
Halmaz	Típus	
a∈Z	a:Egész	a:Z
Külön: Be, Ki	Egyben: A (az állapottér)	
Be: a∈Z Ki: b∈Z	A=(a:Egész,b:Egész)	A=(a:Z, b:Z)
Sorozat: a∈Z[15]	Tömb: a:Tömb[15:Egész]	a:Z[15]
Direkt szorzat: a∈Pont, Pont=x:N x y:N	Rekord: a:Pont, Pont=Rekord(x:Egész, y:Egész)	
Ef: -	Ef=(a=a')	
Uf: (ha a bemenet nem változik)	Uf=(Ef és) (ha a bemenet nem változik)	

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

Állapottér-specifikáció:

```
A =(n:N, jegyek:N[1..n], bukott:L)
Ef=(n=n' és jegyek=jegyek' és
∀i∈[1..n]:(1<=jegyek[i]<=5))
Uf=(Ef és bukott=VAN(i=1..n, jegyek[i]=1))</pre>
```

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

Eldöntés:

```
A =(e:Z, u:Z, van:L)

Ef=(e=e' és u=u')

Uf=(Ef és van=VAN(i=e..u, T(i)))
```

Bukott-e

Be: <u>n∈N</u>, jegyek∈N[1..n]

Ki: bukott∈L

Ef: $\forall i \in [1..n]: (1 <= jegyek[i] <= 5)$

Uf: bukott=VAN(i=1..n,jegyek[i]=1)

- 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
 - vesztettünk-e

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

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

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

- Tippelés
 - a tippelt karakterre nem tippelhettünk korábban

Hibak szama: 6/9

aestkrbnlzp

Tipp:

a tippelt karakter a tömb végére kerül

```
A =(szó:S, n:N, tippek:K[1..n], tipp:K)
   Ef=(szó=szó' és n=n' és tippek=tippek' és tipp=tipp' és
        \forall i \in [1..n]: (tipp \neq tippek[i])
   Uf=(szó=szó' és tipp=tipp' és
        \forall i \in [1..n']: (tippek[i]=tippek'[i])
        n=n'+1 és tippek[n]=tipp)
static void tippeles(string szo, List<char> tippek,
                     char tipp) {
 tippek.Add(tipp);
```

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

```
A =(szó:S, n:N, tippek:K[1..n], i:N, van:L)

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

Uf=(Ef és van=VAN(j=1..n, szó[i]=tippek[j]))
```

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

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