

1. táblás gyakorlat – specifikáció

Specifikáljuk az alábbi feladatot:

Növeljük meg egy számot eggyel!

Első megoldás

A feladat megfogalmazása azt sugallja, hogy mindössze egy változónk van, ami így bemenő és kimenő változóként is szolgál, és az utófeltételben azt szeretnénk elérni, hogy ez a változó az eredeti értékénél eggyel nagyobb legyen (*helyben növelés*).

Azaz:

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

$$ef = (a = a')$$

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

Az utófeltételben nem tarthattuk meg az előfeltételt, hiszen pont az volt a lényeg, hogy a értéke változzon.

Második megoldás

Ha kicsit másképp értelmezzük a feladatot (bár úgy természetes és logikus, ahogy az imént tettük), akkor esetleg úgy is felfoghatjuk, hogy egy adott bemenő változó értékénél eggyel nagyobbat pakoljuk bele egy kimenő változóba, de a bemenő változót hagyjuk békén.

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

$$ef = (a = a')$$

$$uf = (ef \wedge b = a + 1)$$

Itt fontos megjegyezni azt, hogy kezdetben nyilván csak az a értékét „mentettük el”, hiszen a b -ben bármi is lehet, az ezt a feladatot megoldó program ugyanis felülírja majd annak értékét, ezért szerepel az előfeltételben csak az a .

Az utófeltétel szerepe pedig még mindig az, hogy kapcsolatot teremtsen a bemenő szemantikájú változók (*kezdeti*) és a kimenő változók *programfutás utáni* értéke között, azaz itt kötjük ki, hogy a program végrehajtása után b értéke legyen $a+1$. Azért tehetjük meg, hogy nem az a változó programfuttatás előtti értékére hivatkozunk, mert megtartottuk az előfeltételt, azaz azt, hogy a megegyezik a futtatás előtti állapotával. Ezért itt most mindegy, hogy $b=a+1$ -et, vagy $b=a'+1$ -et írunk.

Egy kifejezetten rossz és egy nem túl szerencsés megoldás ugyanerre:

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

$$ef = (a = a')$$

$$uf = (b = a + 1)$$

A fenti, pirossal írt változat helytelen, hiszen az $a, b := 0, 1$ értékadás meg fog felelni az utófeltételnek, annak ellenére, hogy ha mondjuk a kezdeti értéke 6 volt, azt várnánk, hogy b 7 legyen.

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

$$ef = (a = a')$$

$$uf = (b = a' + 1)$$

Ez a verzió már jó, viszont általában, ha megtehetjük, szeretjük megtartani az előfeltételt az utófeltételben. Ennek két oka is van.

Az egyik az, hogy *ha* ragaszkodunk ahhoz a fajta modellhez, hogy vannak tisztán bemenő (itt a) és tisztán kimenő (itt b) értelmezésű változóink, akkor azzal, ha kikötjük, hogy a értéke maradjon ugyanaz a program futása után, mint ami eredetileg volt, még egyértelműbbé tesszük, hogy a egy bemenő szemantikájú változó, hiszen csak „olvasásra” használjuk, nem módosul az értéke.

A másik indok pedig az, hogy mivel a' nem egy állapottérbeli változó, hanem csak egy konstans, ezért őt nem használhatom majd a struktogramban. Ha nekem az algoritmusban valamiért a eredeti értékére van szükségem, akkor vagy korábban ki kellett mentenem egy segédváltozóba, vagy végig kell gondolnom, hogy az algoritmus adott pontján vajon a értéke megváltozott-e és ha igen, mennyivel. Ha kikötöm, hogy a értéke nem változik, akkor a -val könnyebben el lehet bánni a majdani megoldó programban.

Mindez a végiggondolás persze akkor fontos csak, ha mi találjuk ki a specifikációt és az algoritmust. Mi most csak *használni* fogjuk a programozási tételeket, azokban pedig az előfeltételt rendszeresen meg fogjuk tartani.