

Algebrai Axiómák – X-elős, kifejtős megoldások

2008.05.27 – 8. Feladat

Jellemezzünk egy stringet az alábbi műveletekkel! A string karakterei előlről 1-gyel kezdődően számozottak. Adja meg a PAR műveletre vonatkozó algebrai axiómákat! Az axiómák felírásakor használhatja a két egész összehasonlítására és két karakter összehasonlítására szolgáló műveleteket.

CRT ()	<i>konstruktor</i>	új (üres) stringet hoz létre.
SET (s, x)	<i>konstruktor</i>	az s string elejére – az 1. számú helyre – rakja az x karaktert.
PAR (s)	<i>behavior</i>	igaz, ha bárhol a stringben egymás mellett legalább két egyforma karakter áll.
IN (s, i)	<i>behavior</i>	eredményül adja az s string i-ik karakterét. Ha i nagyobb mint a string hossza, akkor az eredmény értelmetlen (nem definiált)./* Itt feltételezve, hogy az $i > 0$, különben az $i < 1$ -re is szűrni kellene */
LGTH (s)	<i>behavior</i>	az s string karaktereinek számát adja.

PAR(CRT()) = false	<i>Triviális hamis, hiszen üres a string.</i>
PAR(SET(s,x)) = if (LGTH(s) == 0) false else if (x == IN(s,1) PAR(s)) true vagy PAR(SET(s,x)) = (LGTH(s) != 0) && (x == IN(s,1) PAR(s))	<i>Ha a string üres, triviálisan hamis, ha nem, akkor pedig akkor lesz igaz, ha az eredeti stringben (s) az 1. karakter az, amit be akarunk szűrni(x), vagy akkor, ha már a stringre alából igaz, hogy PAR(s).</i>
IN(CRT(),i) = undefined	<i>Üres stringben bármilyen i-t is adunk meg, az eredmény értelmetlen, nem definiált.</i>
IN(SET(s,x),i) = if (i > LGTH(s) + 1) undefined else if (i == 1) x else IN(s,i-1)	<i>Ha i nagyobb, mint az új string hossza, akkor a szöveg alapján undefined, ha 1, akkor az újonnan beszúrt x karaktert kell, visszaadja, más különben ($i \leq \text{LGTH}(s)+1$ && $i \neq 1$) az eredeti stringre húzzuk rá az $\text{IN}(s, i-1)$-t.</i>
LGTH(CRT()) = 0	<i>Üres string karaktereinek száma triviálisan 0.</i>
LGTH(SET(s,x)) = LGTH(s) + 1	<i>Ha beszúrunk egy elemet, akkor az eredeti string mérete 1-gyel kell, hogy nőjön.</i>

2008.06. 10 – 8. Feladat

Jellemezzünk egy stringet az alábbi műveletekkel! Adja meg az OCC műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két karakter összehasonlítására szolgáló műveleteket, és az integereken értelmezett alapl műveleteket.

ADD (s, x)	<i>konstruktor</i>	az s string végére rakja az x karaktert.
TAIL (s, n)	<i>modifier</i>	az s string legrégebbi n karakterének levágása után maradó stringet adja. <i>/* Feltételezve, hogy n>0, és ha n nem kisebb, mint a string hossza, akkor üres stringet kapunk */</i>
LGTH (s)	<i>behavior</i>	az s string karaktereinek számát adja.
CRT ()	<i>konstruktor</i>	új (üres) stringet hoz létre.
OCC (s, x)	<i>behavior</i>	az s stringben található x karakterek előfordulásának számát adja.

OCC (CRT (), x) = 0	<i>Triviálisan 0.</i>
OCC (ADD (s, x), y) = if (x == y) OCC (s, y)+1 else OCC (s, y)	<i>Ha a beszúrt elem egyezik a keresett elemmel, akkor az 1-gyel több lesz, mint az eredeti stringben, különben változatlan</i>
TAIL (CRT (), n) = CRT ()	<i>n-től függetlenül egy üres stringet kapunk.</i>
TAIL (ADD (s, x), n) = if (n > LGTH (s)) CRT () else if (n == LGTH (s)) ADD (CRT (), x) else ADD (TAIL (s, n), x)	<i>Ha n nagyobb, mint az eredeti string hossza, akkor a szöveg alapján üres stringet kapunk, ha egyenlő, akkor az annak felel meg, hogy egy üres stringhez adjuk hozzá az x karaktert, minden más esetben (n < LGTH (s)) a sorrend mindegy (először levágjuk az n karaktert, majd a string végére rakjuk x-t).</i>
LGTH (CRT ()) = 0	<i>Triviálisan 0.</i>
LGTH (ADD (s, x)) = LGTH (s) + 1	<i>A beszúrt elemtől triviálisan egyel nő a string mérete.</i>

2009.05.28 – 9. Feladat

Jellemezzünk egy stringet az alábbi műveletekkel! Adja meg a PALIN műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja az egészek és a karakterek összehasonlítására szolgáló műveleteket. Egy string palindróma, ha az elejétől olvasva ugyanaz, mint visszafelé. Pl.: "görög", "abba".

CRT ()	<i>konstruktor</i>	új (üres) stringet hoz létre
LGTH (s)	<i>behavior</i>	az s string karaktereinek számát adja
TAIL (s)	<i>modifier</i>	az s string első karakterének levágása után maradó stringet adja
APPEND (s , x)	<i>konstruktor</i>	az s string végére rakja az x karaktert
HEAD (s)	<i>behavior</i>	az s string első karakterét mutatja meg
PALIN (s)	<i>behavior</i>	igaz, ha az s string palindróma

PALIN (CRT ()) = true	<i>Üres string visszafelé is üres string, true.</i>
PALIN (APPEND (s , x) = LGTH (s) == 0 (PALIN (TAIL (s)) && (HEAD (s) == x)) vagy PALIN (APPEND (s , x) = if (LGTH (s) == 0) true else if (PALIN (TAIL (s)) && HEAD (s) == x) true else false	<i>Ha üres az eredeti string, akkor biztos hogy igaz, hiszen 1 karakter önmagában palindrom, ha pedig nem üres, akkor úgy lehet palindrom, ha az első karakter nélkül a string palindrom E S a beszúrt karakter egyezik az 1. karakterrel. Minden más esetben hamis.</i>
HEAD (CRT ()) = undefined	<i>Üres a string, nincs mit visszaadni.</i>
HEAD (APPEND (s , x) = if (LGTH (s) == 0) x else HEAD (s)	<i>Ha üres a string, akkor logikus, hogy a beszúrt karakter lesz az eredmény, ha nem, akkor meg nem számít a végeredményben a beszúrás, így egyenértékű lesz a HEAD (s)-sel.</i>
TAIL (CRT ()) = CRT ()	<i>Jobb híján a „CRT () CRT () marad”.</i>
TAIL (APPEND (s , x) = if (LGTH (s) == 0) CRT () else APPEND (TAIL (s) , x)	<i>Ha üres a string, beszúrunk, majd törölünk, így egy üres stringet kapunk, ha nem, akkor pedig mindegy a sorrend.</i>
LGTH (CRT ()) = 0	<i>Triviálisan 0.</i>
LGTH (APPEND (s , x) = LGTH (s) + 1	<i>Triviálisan megnő a string mérete 1-gyel.</i>

2009.06.18 – 6. Feladat

Jellemezzünk egy stringet az alábbi műveletekkel! Adja meg a CUT műveletre vonatkozó algebrai axiómákat (a többi axióma nem kell!) ! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket.

CRT ()	<i>konstruktor</i>	új (üres) stringet hoz létre.
SET (s, x)	<i>konstruktor</i>	az s string elejére rakja az x karaktert.
LGTH (s)	<i>behavior</i>	az s string karaktereinek számát adja.
CUT (s, n)	<i>modifier</i>	az s string legrégebbi n darab karakterének levágása után maradó stringet adja. Ha n nem kisebb, mint s string hossza, üres stringet kapunk. Tételjeze fel, hogy $n > 0$!

CUT (CRT (), n) = CRT ()	<i>Feladat szövege alapján triviális.</i>
<pre> CUT (SET (s, x), n) if (n > LGTH(s)) CRT () else if (n == LGTH(s)) SET (CRT (), x) else SET (CUT (s, n), x) </pre>	<i>Nyilván, ha n legalább LGTH (s) + 1, akkor a visszakapott string üres. Ha n az eredeti string méretével egyenlő, akkor az x-t egy üres stringbe rakjuk bele. Minden más esetben (n < LGTH (s)) a sorrend felcserélhető (az egyik az elejére szúr be, a másik a végéről vág le).</i>
LGTH (CRT ()) = 0	<i>Triviális.</i>
LGTH (SET (s, x)) = LGTH (s) + 1	<i>Triviális.</i>

2010.01.12 (B) – 10. Feladat

Jellemezzünk egy (x,y) párokból álló listát – ahol x egy kulcs, y egy 0-nál nagyobb egész érték – az alábbi műveletekkel! Adja meg az algebrai axiómákat! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket.

NEW()	<i>konstruktor</i>	új (üres) listát hoz létre.
ADD(l, (x,y))	<i>konstruktor</i>	az l listához kapcsolja az x,y párt, ha x nem szerepelt a listán. Ha a listán már van x , akkor a hozzá tartozó értéket az új y -nal helyettesíti, ha az nagyobb a listán szereplőnél.
VALUE(l, x)	<i>behavior</i>	megadja az l listán az x kulcshoz tartozó y -t. Ha a listán a megadott x nem szerepel, akkor az eredmény 0.
MAX(l)	<i>behavior</i>	a listában szereplő legnagyobb y . (üres lista esetén nulla)

VALUE(NEW()) = 0	<i>Szöveg alapján triviálisan 0.</i>
VALUE(ADD(l, (x,y), z) = if (x == z && VALUE(l,x) < y) y else VALUE(l,z)	<i>Ha a keresett kulcs megegyezik a beszúrás kulcsával, és a listában hozzátartozó y kisebb, mint az új y, akkor egyértelmű, hogy az új y-ra lecseréljük a régit, így x-hez ez fog tartozni. Ha nem egyezik, akkor visszavezethető a $VALUE(l,z)$-re (mintha be se szúrtuk volna (x,y)-t).</i>
MAX(NEW()) = 0	<i>Szöveg alapján triviális.</i>
MAX(ADD(l, (x,y))) = if (y > MAX(l)) y else MAX(l)	<i>Ha az y nagyobb, mint a listában lévő legnagyobb y, így logikus, hogy y lesz a válasz, ha meg nem, akkor az nem befolyásolja a lista maximumát, marad $MAX(l)$.</i>

2012.01.17 – 7. Feladat

Jellemezzünk egy (K, V) párokból álló Map-et – ahol K egy kulcs, V egy 0-nál nagyobb egész érték – az alábbi műveletekkel! Adja meg az algebrai axiómákat! Az axiómák felírásakor használhatja a két egész összehasonlítására szolgáló műveleteket.

NEW ()	<i>konstruktor</i>	új (üres) map-et hoz létre.
PUT (m, (k, v))	<i>konstruktor</i>	az m map-hez hozzáveszi a k,v párt, ha k nem szerepelt a map-ben. Ha a map-ben már van k , akkor a hozzá tartozó v -t az új v -vel helyettesíti, ha az kisebb a map-ben szereplőnél.
GET (m, k)	<i>behavior</i>	megadja az m map-ben a k kulcshoz tartozó v -t. Ha a map-ben a megadott k nem szerepel, akkor az eredmény 0.
MIN (m)	<i>behavior</i>	az m map-ben szereplő legkisebb v . (üres map esetén nulla)

GET (NEW (), k) = 0	<i>Triviális, ld. feladat szövege.</i>
<pre> GET (PUT (m, (k1, v)), k2) = if (k1 == k2 && (GET (m, k1) == 0 GET (m, k1) < v)) v else GET (m, k2) </pre>	<i>Ha a keresett és beszűrt kulcs megegyezik, ÉS a beszűrt kulcs vagy nem szerepel a listában VAGY nagyobb érték tartozik hozzá, mint v, akkor a k2 kulcshoz a v érték fog tartozni, ha pedig nem egyezik a keresett kulcs a beszűrt kulccsal, akkor mintha se sem szűrtünk volna, visszavezetjük GET (m, k2) -ra.</i>
MIN (NEW ()) = 0	<i>Triviális, ld. feladat szövege.</i>
<pre> MIN (PUT (m, (k, v))) = if (MIN (m) == 0 v < MIN (m)) v else MIN (m) </pre>	<i>Ha a map üres, VAGY a v kisebb, mint a map legkisebb eleme, akkor nyilván ő [v] lesz a legkisebb elem, ha pedig nem, akkor mintha be se szűrtük volna, „marad a map legkisebb eleme a régi”.</i>

2012.12.18 – 7. Feladat

Jellemezzünk egy stringet az alábbi műveletekkel! Egy string palindróma, ha az elejéről olvasva ugyanaz, mint visszafelé. Pl.: "görög", "abba".

CRT ()	<i>konstruktor</i>	új (üres) stringet hoz létre
LGTH (s)	<i>behavior</i>	az s string karaktereinek számát adja
TAIL (s)	<i>modifier</i>	az s string első karakterének levágása után maradó stringet adja
XTEND (s, x)	<i>konstruktor</i>	az s string végére rakja az x karaktert
TOP (s)	<i>behavior</i>	az s string első karakterét mutatja meg
PALIN (s)	<i>behavior</i>	igaz, ha az s string palindróma

Az alábbi táblázatban található kifejezések közül jelölje meg azokat, amelyek algebrai axiómák BAL oldalán állhatnak!

<i>Behaviorban modifier</i>	TOP (TAIL (s))	<i>Behaviorban behavior</i>	PALIN (TAIL (CRT ()))
<i>Behaviorban behavior</i>	LGTH (TOP (CRT ()))	<i>Behaviorban modifier</i>	LGTH (TAIL (s))
<i>Konstruktor van kívül</i>	XTEND (TOP (s))	<i>Behaviorban konstruktor jó paraméterezéssel.</i>	PALIN (XTEND (s, x))
<i>Modifierben konstruktor, de rossz paraméterezéssel. Helyesen: TAIL (XTEND (s, x)).</i>	TAIL (XTEND (CRT ()))	<i>Behaviorban behavior</i>	LGTH (TOP (s))
<i>Behaviorban konstruktor, jó paraméterezéssel.</i>	PALIN (CRT ())	<i>Behaviorban konstruktor jó paraméterezéssel.</i>	TOP (XTEND (s, x))

2013.01.08 – 3. Feladat

Jellemezzünk egy stringet az alábbi műveletekkel!

ADD (s, x)	<i>konstruktor</i>	az s string végére rakja az x karaktert.
LGTH (s)	<i>behavior</i>	az s string karaktereinek számát adja.
NEW ()	<i>konstruktor</i>	új (üres) stringet hoz létre.
OCN (s, x)	<i>behavior</i>	az s stringben található x karakterek előfordulásának számát adja.
TAIL (s)	<i>modifier</i>	az s string legrégebbi karakterének levágása után maradó stringet adja.

Az alábbi kifejezésekhez adja meg, hogy azok algebrai axiómák BAL oldalán állhatnak-e vagy sem!

<i>Behaviorban modifier</i>	OCN (TAIL (s), x)	<i>Konstruktorban konstruktor</i>	ADD (NEW (), x)
<i>Behaviorban behavior</i>	LGTH (OCN (s, x))	<i>Behaviorban konstruktor, jó paraméterezéssel.</i>	OCN (ADD (s, y), x)
<i>Behaviorban konstruktor, de rossz paraméterezéssel. Helyesen: (OCN (ADD (s, x), y)).</i>	OCN (ADD (s, x), x)	<i>Behaviorban konstruktor, de rossz paraméterezéssel. Helyesen: OCN (NEW (), x).</i>	OCN (NEW (), 0)
<i>Modifierben behavior</i>	TAIL (OCN (s, x))	<i>Behaviorban modifier</i>	LGTH (TAIL (s))
<i>Behaviorban konstruktor, de rossz paraméterezéssel. Helyesen: LGTH (ADD (s, x)).</i>	LGTH (ADD (NEW (), x))	<i>Konstruktor van kívül.</i>	ADD (s, LGTH (s))

2013.06.11 – 7. Feladat

Jellemezzünk egy stringet az alábbi műveletekkel!

END (s, x)	<i>konstruktor</i>	az s string végére rakja az x karaktert.
CHR (s, x)	<i>behavior</i>	az s stringben található x karakterek előfordulásának számát adja.
LGTH (s)	<i>behavior</i>	az s string karaktereinek számát adja.
CRT ()	<i>konstruktor</i>	új (üres) stringet hoz létre.
TAIL (s)	<i>modifier</i>	az s string legrégebbi karakterének levágása után maradó stringet adja.

Az alábbi kifejezésekhez adja meg, hogy azok algebrai axiómák BAL oldalán állhatnak-e vagy sem!

<i>Behaviorban modifier</i>	CHR(TAIL(s), x)	<i>Konstruktor van kívül.</i>	END(CRT(), x)
<i>Modifierben behavior</i>	TAIL(CHR(s, x))	<i>Behaviorban modifier.</i>	LGTH(TAIL(s))
<i>Behaviorban konstruktor, rossz paraméterezéssel (CHR(END(s, x), y) lenne a helyes).</i>	CHR(END(s, x), x)	<i>Behaviorban konstruktor, de rossz paraméterezéssel. Helyesen: CHR(CRT(), x)</i>	CHR(CRT(), 0)
<i>Behaviorban konstruktor, jó paraméterezéssel.</i>	CHR(END(s, y), x)	<i>Behaviorban behavior</i>	LGTH(CHR(s, x))
<i>Modifierben behavior</i>	TAIL(CHR(s, x))	<i>Behaviorban modifier</i>	LGTH(TAIL(s))
<i>Behaviorban konstruktor, de rossz paraméterezéssel. Helyesen: LGTH(END(s, x)).</i>	LGTH(END(CRT(), x))	<i>Konstruktorral kezdődik.</i>	END(s, LGTH(s))

2014.01.14 – 4. Feladat

Jellemezzünk egy maximum 10 elemet tartalmazó halmazt az alábbi műveletekkel!

CRT ()	<i>konstruktor</i>	új (üres) halmazt hoz létre.
MBR (s, i)	<i>behavior</i>	igaz, ha az i elem az s halmaz eleme.
INS (s, i)	<i>konstruktor</i>	az s halmazhoz adja az i elemet, ha a halmaz elemeinek száma nem nő 10 fölé. Ellenkező esetben a művelet hatástalan.
RMV (s, i)	<i>modifier</i>	az s halmazból az i elem eltávolításával kapott halmazt adja meg. Ha i nem volt a halmazban, akkor művelet hatástalan.
SIZE (s)	<i>behavior</i>	megadja az s halmaz elemeinek számát.

Az alábbi kifejezésekhez adja meg, hogy azok algebrai axiómák BAL oldalán állhatnak-e vagy sem!

<i>Modifierben konstruktor, de rossz paraméterezéssel. Helyesen: RMV (INS (s, x), y).</i>	RMV (INS (s, x), x)	<i>Konstruktor van kívül.</i>	CRT (INS (s, x))
<i>Behaviorban modifier.</i>	MBR (RMV (s, x), y)	<i>Behaviorban behavior.</i>	SIZE (MBR (s, x))
<i>Behaviorban konstruktor, de rossz paraméterezéssel. Helyesen: SIZE (INS (s, i)).</i>	SIZE (INS (CRT ()), y)	<i>Behaviorban konstruktor, de rossz paraméterezéssel. Helyesen: MBR (INS (s, x), x).</i>	MBR (INS (s, x), x)
<i>Konstruktor van kívül.</i>	INS (CRT ()), x)	<i>Konstruktor van kívül.</i>	INS (RMV (s, x), y)
<i>Behaviorban modifier</i>	SIZE (RMV (s, y))	<i>Modifierben konstruktor, jó paraméterezéssel.</i>	RMV (INS (s, y), x)