Anna Plust, 36148
Automatyczna optymalizacja oprogramowania

Laboratorium 1: Kalkulator ISCC, podstawowe operatory na zbiorach i relacjach

Wariant 5.7
Zbiór niesparametryzowany
S:={ [i,j] : 1 <= i <= 12 and 1 <= j <= i };

Zbiór sparametryzowany
S:=[n] ->{ [i,j] : 1 <= i <= n and 1 <= j+12 <= i };

Relacja:
R:=[n] -> { [i,j] -> [i, j + 12] : 0 <= i < n and 0 <= j < n -1 };

Zadanie 4
Wykonać dla wskazanego zbioru niesparametryzowanego i sparametryzowanego oraz relacji operacje niżej.

Zadanie 4.1
S:={ [i,j] : 1 <= i <= 12 and 1 <= j <= i }; print "S:"; S;
$0 := { [i, j] : 0 < i <= 12 and 0 < j <= i }

Zadanie 4.2
S:=[n] ->{ [i,j] : 1 <= i <= n and 1 <= j+12 <= i }; S;
$1 := [n] -> { [i, j] : 0 < i <= n and -11 <= j <= -12 + i }

Zadanie 4.3
R:=[n] -> { [i,j] -> [i, j + 12] : 0 <= i < n and 0 <= j < n -1 }; R;
$2 := [n] -> { [i, j] -> [i, 12 + j] : 0 <= i < n and 0 <= j <= -2 + n }

Zadanie 4.4
CARD – oblicza liczbę elementów zbioru
q := card S; q;
Niesparametryzowany: *$3 := { 78 }*
Sparametryzowany: *$3 := [n] -> { (1/2 * n + 1/2 * n^2) : n > 0 }*

Zadanie 4.5
D := domain R; //obliczenie dziedziny relacji D;
$4 := [n] -> { [i, j] -> [i, 12 + j] : 0 <= i < n and 0 <= j <= -2 + n }

Zadanie 4.6
RA:= range R; //obliczenie zakresu relacji RA;
$5 := [n] -> { [i0, i1] : 0 <= i0 < n and 12 <= i1 <= 10 + n }

Zadanie 4.7
m := identity S; // tworzy relację tożsamości m;
Niesparametryzowany: *$7 := { [i, j] -> [i' = i, j' = j] : 0 < i <= 12 and 0 < j <= i }*
Sparametryzowany: *$8 := [n] -> { [i, j] -> [i' = i, j' = j] : 0 < i <= n and -11 <=*
j <= -12 + i }

Zadanie 4.8
I := pow R; // oblicza potęgę relacji m

```
$9 := ([n] -> { [k] -> [[i, j] -> [i, 12k + j]] : k > 0 and 0 <= i < n and 0 <= j <= 10 +
n - 12k }, True)
```

Zadanie 4.9

S2 := scan S; // przebiera elementy skończonego zbioru – zbiór musi być niesparametryzowany
S2;

Niesparametryzowany: `$10 := { [i = 12, j = 12]; [i = 12, j = 11]; [i = 11, j = 11]; [i`
`= 12, j = 10]; [i = 11, j = 10]; [i = 10, j = 10]; [i = 12, j = 9]; [i = 11, j = 9]; [i =`
`10, j = 9]; [i = 9, j = 9]; [i = 12, j = 8]; [i = 11, j = 8]; [i = 10, j = 8]; [i = 9, j`
`= 8]; [i = 8, j = 8]; [i = 12, j = 7]; [i = 11, j = 7]; [i = 10, j = 7]; [i = 9, j = 7];`
`[i = 8, j = 7]; [i = 7, j = 7]; [i = 12, j = 6]; [i = 11, j = 6]; [i = 10, j = 6]; [i =`
`9, j = 6]; [i = 8, j = 6]; [i = 7, j = 6]; [i = 6, j = 6]; [i = 12, j = 5]; [i = 11, j =`
`5]; [i = 10, j = 5]; [i = 9, j = 5]; [i = 8, j = 5]; [i = 7, j = 5]; [i = 6, j = 5]; [i =`
`5, j = 5]; [i = 12, j = 4]; [i = 11, j = 4]; [i = 10, j = 4]; [i = 9, j = 4]; [i = 8, j =`
`4]; [i = 7, j = 4]; [i = 6, j = 4]; [i = 5, j = 4]; [i = 4, j = 4]; [i = 12, j = 3]; [i =`
`11, j = 3]; [i = 10, j = 3]; [i = 9, j = 3]; [i = 8, j = 3]; [i = 7, j = 3]; [i = 6, j =`
`3]; [i = 5, j = 3]; [i = 4, j = 3]; [i = 3, j = 3]; [i = 12, j = 2]; [i = 11, j = 2]; [i`
`= 10, j = 2]; [i = 9, j = 2]; [i = 8, j = 2]; [i = 7, j = 2]; [i = 6, j = 2]; [i = 5, j =`
`2]; [i = 4, j = 2]; [i = 3, j = 2]; [i = 2, j = 2]; [i = 12, j = 1]; [i = 11, j = 1]; [i`
`= 10, j = 1]; [i = 9, j = 1]; [i = 8, j = 1]; [i = 7, j = 1]; [i = 6, j = 1]; [i = 5, j =`
`1]; [i = 4, j = 1]; [i = 3, j = 1]; [i = 2, j = 1]; [i = 1, j = 1] }`

Zadanie 4.10

R2 := scan(R * [n] -> {:n=6}); // przebiera elementy skończonej relacji, relacja musi być niesparametryzowana
R2;

`$11 := [n] -> { [i = 5, j = 4] -> [5, 16] : n = 6; [i = 4, j = 4] -> [4, 16] : n = 6; [i`
`= 3, j = 4] -> [3, 16] : n = 6; [i = 2, j = 4] -> [2, 16] : n = 6; [i = 1, j = 4] -> [1,`
`16] : n = 6; [i = 0, j = 4] -> [0, 16] : n = 6; [i = 5, j = 3] -> [5, 15] : n = 6; [i =`
`4, j = 3] -> [4, 15] : n = 6; [i = 3, j = 3] -> [3, 15] : n = 6; [i = 2, j = 3] -> [2,`
`15] : n = 6; [i = 1, j = 3] -> [1, 15] : n = 6; [i = 0, j = 3] -> [0, 15] : n = 6; [i =`
`5, j = 2] -> [5, 14] : n = 6; [i = 4, j = 2] -> [4, 14] : n = 6; [i = 3, j = 2] -> [3,`
`14] : n = 6; [i = 2, j = 2] -> [2, 14] : n = 6; [i = 1, j = 2] -> [1, 14] : n = 6; [i =`
`0, j = 2] -> [0, 14] : n = 6; [i = 5, j = 1] -> [5, 13] : n = 6; [i = 4, j = 1] -> [4,`
`13] : n = 6; [i = 3, j = 1] -> [3, 13] : n = 6; [i = 2, j = 1] -> [2, 13] : n = 6; [i =`
`1, j = 1] -> [1, 13] : n = 6; [i = 0, j = 1] -> [0, 13] : n = 6; [i = 5, j = 0] -> [5,`
`12] : n = 6; [i = 4, j = 0] -> [4, 12] : n = 6; [i = 3, j = 0] -> [3, 12] : n = 6; [i =`
`2, j = 0] -> [2, 12] : n = 6; [i = 1, j = 0] -> [1, 12] : n = 6; [i = 0, j = 0] -> [0,`
`12] : n = 6 }`

Zadanie 4.11

`S1:={ [i,j] : 1 <= i <= 3 and 1 <= j <= i };`
S3 := S + S1; // suma zbiorów, zbiór S1 utworzyć samemu
S3;
`$12 := { [i, j] : i <= 12 and 0 < j <= i }`

Zadanie 4.12

`R1:=[n] -> { [i,j] -> [i+1, j] : 0 <= i < n and 0 <= j < n - 1 };`
R3:= R+R1; // suma relacji, relację R1 utworzyć samemu
R3;
`$13 := [n] -> { [i, j] -> [i, 12 + j] : 0 <= i < n and 0 <= j <= -2 + n; [i, j] -> [1 +`
`i, j] : 0 <= i < n and 0 <= j <= -2 + n }`

Zadanie 4.13

S3:= S – S1; //różnica zbiorów
$22 : = { };

Zadanie 4.14
R3:= R – R1; //różnica relacji
$27 := [n] -> { [i, j] -> [i, 12 + j] : 0 <= i < n and 0 <= j <= -2 + n }

Zadanie 4.15
S3 := S ∗ S1; // intersekcja zbiorów
$23 := { [i, j] : i <= 12 and 0 < j <= i }

Zadanie 4.16
R2 := R∗S // intersekcja dziedziny relacji R ze zbiorem S
$24 := [n] -> { [i, j] -> [i, 12 + j] : 0 <= i <= 12 and i < n and 0 < j <= i and j <= -2 + n }

Zadanie 4.17
R2 := R(S); // aplikacja relacji S na zbiorze S
$15 := [n] -> { [i0, i1] : 0 <= i0 <= 12 and i0 < n and 13 <= i1 <= 12 + i0 and i1 <= 10 + n }

Zadanie 4.18
R3 := R. R1; // kompozycja relacji R i R1
$16 := [n] -> { [i, j] -> [1 + i, 12 + j] : 0 <= i < n and 0 <= j <= -14 + n }

Zadanie 4.19
R2 := R^i; //relacja R do potęgi i, gdzie i jest stała
i := 2;
$25 := [n] -> { [i, j] -> [i, 24 + j] : 0 <= i < n and 0 <= j <= -14 + n }

Zadanie 4.20
I := R^+; // obliczenie tranzytywnego domknięcia relacji R
$23 := ([n] -> { [i, j] -> [i, o1] : (-j + o1) mod 12 = 0 and 0 <= i < n and 0 <= j <= -2 + n and o1 >= 12 + j and 12 <= o1 <= 10 + n }, True)

Zadanie 4.21
b := S= S1; //sprawdzenie czy zbiory S i S1 są identyczne
$17 := False

Zadanie 4.22
b := R= R1; //sprawdzenie czy relacje R i R1 są identyczne
$18 := False

Zadanie 4.23
b := S<= S1; //sprawdzenie czy zbiór S jest podzbiorem S1
$19 := False

Zadanie 4.24
b := R<= R1; //sprawdzenie czy relacja R jest podzbiorem R1
$20 := False

Zadanie 4.25

m := S<< S; //tworzenie relacji, której elementy dziedziny są // leksykograficznie ściśle mniejsze niż elementy zakresu.

$21 := { [i, j] -> [i', j'] : i <= 12 and 0 < j <= i and i < i' <= 12 and 0 < j' <= i'; [i, j] -> [i' = i, j'] : i <= 12 and 0 < j <= i and j' > j and 0 < j' <= i }

Zadanie 4.26

m := S >> S; //tworzenie relacji, dla której elementy dziedziny są // leksykograficznie ściśle większe niż elementy zakresu.

$22 := { [i, j] -> [i', j'] : i <= 12 and 0 < j <= i and i' <= 12 and i' < i and 0 < j' <= i'; [i, j] -> [i' = i, j'] : i <= 12 and 0 < j <= i and 0 < j' < j and j' <= i }