Patryk Wójtowicz

Tablica podsumowująca testy pojedyńczych relacji:

Relacja	Liczba testów
Follows	4
Follows*	3
Parent	3
Parent*	2
Modifies	7
Uses	7
Calls	4

Tablica podsumowująca testy dwóch lub więcej relacji:

Relacje	Liczba testów
Modifies, Uses	12
Follows, Uses	5
Calls, Modifies	8
Uses, Calls	2
Follows, Modifies	3

Testy Pojedyńcze:

stmt s, s1;

Select s such that Follows* (s, s1)

1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 17, 18, 19, 20, 21, 22, 24, 27, 23, 30, 31, 32, 29, 35, 36, 39, 40, 41, 43, 34, 45, 46, 48, 49, 50, 47, 54, 56, 55, 60, 61, 16, 63, 64, 65, 67, 68, 70, 71, 72, 81, 82, 84, 85, 90, 91, 93, 14, 96, 97, 95, 102, 101, 106, 108, 110, 111, 114, 113, 12, 6, 120, 121, 122, 123, 124, 125, 126, 127, 128, 130, 132, 133, 134, 135, 137, 138, 139, 140, 145, 147, 143, 150, 151, 152, 153, 154, 155, 156, 157, 161, 162, 163, 166, 169, 170, 182, 183, 184, 186, 188, 190, 193, 192, 197, 199, 201, 198, 196, 204, 191, 210, 211, 213, 214, 219, 220, 221, 222, 223, 224, 227, 228, 229, 230, 218, 234, 244, 245, 246, 247, 248, 249, 252, 251, 255, 257, 250, 259, 261, 263, 267, 270, 271, 272, 266, 275, 265, 283, 284, 285, 286, 290, 295, 296, 298, 299, 300, 302, 303, 306, 307

assign a;

Select a such that Uses (a, "l")

13, 21, 39, 61, 65, 70, 71, 75, 77, 91, 104, 252, 253, 254

assign a1;

Select a1 such that Modifies (a1, "width")

2, 30, 49, 82, 85

stmt s1; while w;

Select w such that Parent* (w, s1)

6, 12, 16, 26, 29, 47, 59, 69, 79, 83, 89, 95, 101, 103, 105, 113, 136, 143, 180, 181, 184, 187, 191, 196, 209, 217, 218, 234, 239, 251, 256, 264, 265, 279, 281, 289, 301

stmt s; if ifstat;

Select ifstat such that Follows (ifstat, s)

23, 34, 55, 72, 14, 97, 140, 163, 166, 170, 192, 198, 204, 224, 230, 250, 267, 266, 303

procedure p;

Select p such that Calls (p, "Init")

Main

variable v;

Select v such that Uses ("Main", v)

width, height, tmp, l, x1, incre, left, x2, right, y1, top, y2, bottom, area, j, difference, k, decrement, radius, x3, x4, volume, length, x5, x8, x9, x6, x, c, weight, factor, pct, mtoggle, dx, lengx, cover, dy, marking, median, asterick, range, mean, s, pink, green, pixel, dot, notmove, line, edge, depth, semi, increase, temporary, decrease, half, notdone, triangle, base, degrees, triange

constant c; variable v;

Select c such that Uses ("Random", v)

1, 0, 10, 3, 2, 16, 83, 11, 32, 5, 100, 8, 1000, 20

constant c; variable v;

Select c such that Modifies ("Transform", v)

1, 0, 10, 3, 2, 16, 83, 11, 32, 5, 100, 8, 1000, 20

stmt s1; prog_line n;

Select n such that Parent (n, s1)

6, 12, 14, 15, 16, 23, 26, 29, 34, 38, 47, 51, 55, 59, 66, 69, 72, 76, 79, 80, 83, 86, 89, 95, 97, 101, 103, 105, 107, 109, 113, 136, 140, 143, 144, 159, 160, 163, 166, 170, 173, 176, 180, 181, 184, 187, 191, 192, 196, 198, 204, 209, 216, 217, 218, 224, 230, 234, 237, 239, 241, 250, 251, 256, 264, 265, 266, 267, 278, 279, 281, 289, 291, 301, 303, 309

prog_line n1, n2;

Select n1 such that Follows* (n1, n2)

1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 17, 18, 19, 20, 21, 22, 24, 27, 23, 30, 31, 32, 29, 35, 36, 39, 40, 41, 43, 34, 45, 46, 48, 49, 50, 47, 54, 56, 55, 60, 61, 16, 63, 64, 65, 67, 68, 70, 71, 72, 81, 82, 84, 85, 90, 91, 93, 14, 96, 97, 95, 102, 101, 106, 108, 110, 111, 114, 113, 12, 6, 120, 121, 122, 123, 124, 125, 126, 127, 128, 130, 132, 133, 134, 135, 137, 138, 139, 140, 145, 147, 143, 150, 151, 152, 153, 154, 155, 156, 157, 161, 162, 163, 166, 169, 170, 182, 183, 184, 186, 188, 190, 193, 192, 197, 199, 201, 198, 196, 204, 191, 210, 211, 213, 214, 219, 220, 221, 222, 223, 224, 227, 228, 229, 230, 218, 234, 244, 245, 246, 247, 248, 249, 252, 251, 255, 257, 250, 259, 261, 263, 267, 270, 271, 272, 266, 275, 265, 283, 284, 285, 286, 290, 295, 296, 298, 299, 300, 302, 303, 306, 307

prog_line n2;

Select n2 such that Modifies (n2, "tmp")

4, 6, 12, 14, 15, 16, 17, 29, 33, 34, 38, 42, 47, 48, 59, 60, 63, 66, 76, 78, 79, 80, 81, 89, 90, 95, 96, 105, 107, 108, 133, 136, 137, 149, 180, 181, 187, 188

stmt s, s1;

Select s1 such that Follows* (s, s1)

2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14, 18, 19, 20, 21, 22, 23, 25, 28, 29, 31, 32, 33, 34, 36, 37, 40, 41, 42, 44, 45, 46, 47, 49, 50, 51, 54, 55, 57, 59, 61, 62, 63, 64, 65, 66, 68, 69, 71, 72, 75, 82, 83, 85, 86, 91, 92, 94, 95, 97, 100, 101, 103, 105, 107, 109, 111, 112, 115, 116, 118, 119, 121, 122, 123, 124, 125, 126, 127, 128, 129, 131, 133, 134, 135, 136, 138, 139, 140, 143, 146, 148, 149, 151, 152, 153, 154, 155, 156, 157, 158, 162, 163, 166, 169, 170, 173, 183, 184, 186, 187, 189, 191, 194, 196, 198, 200, 202, 203, 204, 207, 208, 211, 212, 214, 215, 220, 221, 222, 223, 224, 227, 228, 229, 230, 233, 234, 236, 245, 246, 247, 248, 249, 250, 253, 254, 256, 258, 259, 260, 262, 264, 270, 271, 272, 273, 275, 276, 277, 284, 285, 286, 287, 291, 296, 297, 299, 300, 301, 303, 306, 307, 308

assign a2;

Select a2 such that Modifies (a2, "height")

3, 31, 61, 88, 91, 272

assign a; while w;

Select w such that Parent (w, a)

6, 12, 16, 26, 29, 47, 59, 69, 83, 89, 95, 101, 103, 105, 113, 136, 181, 184, 187, 196, 191, 209, 218, 217, 239, 251, 256, 265, 279, 281, 289, 301

assign a; if ifstat;

Select ifstat such that Follows (ifstat, a)

72, 97, 166, 198, 204, 230, 267, 266, 303

procedure p;

Select p such that Calls (p, "Random")

Main, Draw, Rotate

variable v;

Select v such that Uses ("Draw", v)

x1, incre, x2, y1, top, y2, bottom, difference, decrement, pct, mtoggle, dx, lengx, cover, dy, marking, median, asterick, range, mean, s, pink, green

constant c; variable v;

Select c such that Uses ("Enlarge", v)

1, 0, 10, 3, 2, 16, 83, 11, 32, 5, 100, 8, 1000, 20

assign a1;

Select a1 such that Modifies (a1, "width")

2, 30, 49, 82, 85

assign a; prog_line n;

Select n such that Parent* (n, a)

6, 12, 14, 15, 16, 23, 26, 29, 34, 38, 47, 51, 55, 59, 66, 69, 72, 76, 79, 80, 83, 86, 89, 95, 97, 101, 103, 105, 107, 109, 113, 136, 140, 143, 144, 159, 160, 163, 166, 170, 173, 176, 180, 181, 184, 187, 191, 192, 196, 198, 204, 209, 216, 217, 218, 224, 237, 239, 241, 250, 251, 256, 264, 265, 266, 267, 278, 279, 281, 289, 301, 303, 309

prog_line n1; if ifstat;

Select n1 such that Follows (n1, ifstat)

13, 22, 29, 50, 54, 65, 71, 85, 96, 106, 108, 139, 162, 163, 169, 170, 197, 196, 223, 229, 249, 290, 302

prog_line n2;

Select n2 such that Modifies (n2, "area")

6, 11, 12, 14, 15, 16, 29, 32

assign a;

Select a such that Uses (a, "l")

13, 21, 39, 61, 65, 70, 71, 75, 77, 91, 104, 252, 253, 254

assign a2;

Select a2 such that Modifies (a2, "volume")

40, 56

while w; if ifstat;

Select w such that Parent (w, ifstat)

12, 16, 47, 69, 79, 83, 95, 105, 136, 143, 191, 196, 218, 265, 289, 301

assign a1; if ifstat;

Select ifstat such that Follows (ifstat, a1)

72, 97, 166, 198, 204, 230, 267, 266, 303

procedure p;

Select p such that Calls (p, "Transform")

Main

variable v;

Select v such that Uses ("Shrink", v)

I, x1, incre, x2, y1, top, y2, bottom, j, difference, decrement, factor, pct, mtoggle, dx, lengx, cover, dy, marking, median, asterick, range, mean, s, pink, green

procedure p;

Select p such that Calls (p, "Transform")

Main

Testy dwóch lub więcej relacji:

variable v;

Select v such that Modifies("Main", v) and Uses("Init", v)

none

prog_line n1, n2;

Select n1 such that Follows(n1, n2) and Uses(n2, "I")

5, 11, 13, 20, 21, 29, 41, 55, 60, 16, 64, 65, 68, 70, 72, 82, 90, 95, 102, 183, 186, 249, 252, 251

procedure p;

Select p such that Calls(p, "Init") and Modifies(p, "x1")

Main

procedure p;

Select p such that Uses(p, "I") and Calls(p, "Random")

Main

variable v;

Select v such that Uses("Shift", v) and Modifies("Transform", v)

x1, incre, x2, y1, y2, decrement

variable v;

Select v such that Modifies("Draw", v) and Uses("Main", v)

left, right, pct, dx, dy, marking, asterick, range, s, pink, green

prog_line n1, n2;

Select n1 such that Follows(n1, n2) and Uses(n2, "x1")

5, 7, 11, 13, 18, 21, 23, 32, 29, 35, 36, 40, 41, 34, 45, 46, 48, 16, 63, 65, 68, 70, 71, 81, 82, 84, 85, 101, 106, 108, 12, 135, 139, 140, 147, 151, 152, 153, 154, 155, 156, 157, 161, 162, 166, 183, 190, 196, 244, 248, 249, 251, 250

prog_line n1, n2;

Select n1 such that Follows(n1, n2) and Modifies(n2, "width")

1, 5, 11, 13, 23, 46, 48, 81, 82, 84

procedure p;

Select p such that Calls(p, "Shear") and Modifies(p, "x1")

Main

variable v;

Select v such that Uses("Enlarge", v) and Modifies("Shrink", v)

pink, green

prog_line n1, n2;

Select n1 such that Modifies(n1, "tmp") and Follows(n1, n2)

4, 6, 12, 14, 16, 17, 29, 34, 47, 48, 60, 63, 81, 90, 95, 96, 108, 133, 137, 188

procedure p;

Select p such that Calls(p, "Move") and Uses(p, "I")

Main

procedure p;

Select p such that Calls(p, "Transform") and Modifies(p, "incre")

Main

variable v;

Select v such that Uses("Translate", v) and Modifies("Rotate", v)

height, dx, pink, green, dot, edge, triangle, base

variable v;

Select v such that Modifies("Fill", v) and Uses("Enlarge", v)

depth, semi

variable v;

Select v such that Uses("Main", v) and Modifies("Init", v)

x1, incre, left, x2, right, y1, top, y2, bottom, decrement

prog_line n1, n2;

Select n1 such that Follows(n1, n2) and Modifies(n2, "radius")

5, 11, 13, 18, 29

variable v;

Select v such that Modifies("Main", v) and Uses("Transform", v)

height, tmp, x1, incre, left, x2, right, y1, top, y2, bottom, decrement, weight

procedure p;

Select p such that Calls(p, "Clear") and Modifies(p, "s")

Draw

variable v;

Select v such that Modifies("Shear", v) and Uses("Shrink", v)

x1, incre, x2, y1, y2, decrement, factor

procedure p;

Select p such that Calls(p, "Random") and Modifies(p, "left")

Main, Draw, Rotate

prog_line n1, n2;

Select n2 such that Follows(n1, n2) and Uses(n2, "area")

6, 12, 14, 29, 34, 95, 105, 107, 109, 112

variable v;

Select v such that Modifies("Enlarge", v) and Uses("Draw", v)

pink, green

procedure p;

Select p such that Calls(p, "Shift") and Modifies(p, "x3")

Main

prog_line n1, n2;

Select n1 such that Follows(n1, n2) and Uses(n2, "I")

5, 11, 13, 20, 21, 29, 41, 55, 60, 16, 64, 65, 68, 70, 72, 82, 90, 95, 102, 183, 186, 249, 252, 251

variable v;

Select v such that Uses("Move", v) and Modifies("Shrink", v)

I, x1, x2, factor

procedure p;

Select p such that Calls(p, "Scale") and Modifies(p, "wcounter")

none

variable v;

Select v such that Uses("Translate", v) and Modifies("Rotate", v)

height, dx, pink, green, dot, edge, triangle, base

prog_line n1, n2;

Select n2 such that Follows(n1, n2) and Uses(n2, "volume")

6, 12, 14, 55

procedure p;

Select p such that Calls(p, "Fill") and Modifies(p, "temporary")

Enlarge

SIMPLE:

```
procedure Main {
 call Init;
 width = 1;
 height = 0;
 tmp = 0;
 call Random;
 while I {
  x1 = width + incre + left;
  x2 = x1 + incre + right;
  y1 = height + incre * top;
  y2 = y1 + incre * bottom;
  area = width * height;
  while j {
   difference = I + j - 10;
   if j then {
    if k then {
     while difference {
      tmp = decrement * area;
      call Enlarge;
      radius = x1 * 3 + difference;
      difference = difference - 1;
      x2 = j + k - I;
      call Shrink;
      if tmp then {
```

```
  y1 = 0;
  y2 = 0; }
else {
  while y1 {
    y1 = y1 - 1;
    y2 = tmp - 1; } }
while area {
  width = x1 * x2 + incre * left;
  height = right - y1 - incre * y2;
  area = width * height;
  call Transform; }
if area then {
  radius = difference + 3 - (2 * incre);
  x3 = radius + x1;
  difference = x4 + x1 - incre; }
else {
  if I then {
    I = I - 1;
    volume = height * 1 - (3 * width);
    call Shear;
    call Move; }
  else {
    distance = length + x1;
    call Random; } }
call Shift;
length = height * x2 - x1;
while length {
  tmp = tmp - 1;
  width = x2 + x1 - left;
  length = y2 - y1 + tmp;
  if length then {
```

```
      length = length * 0; }
    else {
      length = 0; } }
  call Random;
  if volume then {
    volume = x4 + x3 - x5;
    x5 = 16 * (tmp + 83); }
  else {
    x8 = volume * 11 + volume - x9 + volume; }
  while top {
    tmp = 0;
    height = tmp - k + I + y2;
    call Enlarge; } }
call Move;
x5 = x1 + y2 - 3;
incre = I + k - decrement;
if x6 then {
  x1 = top + bottom - difference;
  x6 = x5 + 32;
  while I {
    I = I - 1;
    x6 = x2 + x1 - x3 * I;
    if j then {
      j = j - 1; }
    else {
      x2 = x1 + radius - tmp; }
    I = I - 1; } }
else {
  if k then {
    top = width - I - j; }
  else {
```

```
      call Transform; } } }
  else {
    while difference {
      if incre then {
        tmp = 0;
        width = x2 - x1;
        while width {
          call Shrink;
          width = width - 2 + x1;
          if height then {
            call Draw; }
          else {
            height = 0; } } }
      else {
        while top {
          tmp = 0;
          height = tmp - k + I + y2;
          call Enlarge; } } } } }
else {
  x7 = x8 + y1 - incre;
  y7 = 0; }
while area {
  tmp = 1;
  if tmp then {
    I = 0; }
  else {
    j = 0; }
  j = 0; }
while radius {
  circumference = 1 * radius + tmp;
  while tmp {
```

```
        circumference = I - (k + j * decrement); } }
      while x {
       x = x + 1;
       if left then {
         call Transform;
         if right then {
           incre = incre - 1;
           b = 0;
           c = area + length * width + incre; }
         else {
           while c {
             call Shift;
             c = c - 1; }
           x = x + 1; } }
       else {
         call Translate; } } }
  call Draw; }
  call Init; }

procedure Init {
  x1 = 0;
  x2 = 0;
  y1 = 0;
  y2 = 0;
  left = 1;
  right = 1;
  top = 1;
  bottom = 1;
  incre = 10;
  decrement = 5; }
```

```
procedure Random {
  left = incre * bottom;
  right = decrement * top; }

procedure Transform {
  weight = 1;
  tmp = 100;
  incre = incre * weight;
  decrement = top - bottom + (right - left) * weight;
  while tmp {
    tmp = incre + height * weight;
    x1 = x2 + tmp;
    x2 = tmp * weight - tmp;
    if x2 then {
      weight = y2 - y1; }
    else {
      weight = x2 - x1; }
    while tmp {
      if weight then {
        y2 = y2 + incre;
        y1 = y1 - decrement; }
      else {
        y1 = x2 * tmp;
        y2 = x1 * (height - bottom); } }
    tmp = 0; } }

procedure Shift {
  top = x2 - x1 * incre;
  bottom = y2 * y1 - decrement;
  x3 = x1 + x2 * y1 + y2 * left + right;
  x4 = x1 * x2 + y1 * y2 + left - right;
```

```
  x5 = x1 + x2 + y1 + y2 * (left - right);
  x6 = x1 * x2 * y1 - y2 + left * right;
  x7 = x1 * x2 * y1 * y2 * left * right;
  x8 = (x1 + x2 * y1) * (y2 * left - right);
  x9 = x1 + x2 * y1 * y2 * left - right; }

procedure Shear {
  if x1 then {
    if x2 then {
      y1 = y2 + incre;
      incre = x2 - x1;
      if y1 then {
        x1 = 0; }
      else {
        x1 = decrement + x1; }
      if y2 then {
        x2 = incre * 2; }
      else {
        x2 = y2 - y1; }
      decrement = (x1 + x2) * (y1 + y2);
      if decrement then {
        factor = 0; }
      else {
        factor = 1; }
      if factor then {
        x1 = 0; }
      else {
        x2 = 0; } }
    else {
      if y1 then {
        y1 = 0; }
```

```
      else {
        y1 = y1 - factor; } } }
   else {
    y2 = 0; } }

procedure Move {
  while tmp {
    while factor {
      x1 = x2 + incre * factor;
      factor = factor - 1;
      while I {
        I = x1 + decrement; }
      x2 = tmp * factor - (height * width);
      while I {
        tmp = factor;
        factor = 0; } } } }

procedure Draw {
  call Clear;
  while pct {
    if mtoggle then {
      dx = lengx + 1 - cover * pct;
      dy = dx * marking - median; }
    else {
      call Random; }
    while asterick {
      range = dx - dy + range;
      if range then {
        peak = marking - y2 * mean;
        marking = marking - 1; }
      else {
```

```
      pct = 0;
      trim = 0; }
    range = range + 1; }
    if pct then {
      pct = 0; }
    else {
      asterick = x1 * x1 + y1 * x2; }
  pct = pct - 1; }
 call Show; }

procedure Clear {
 while s {
  p1 = 0;
  p2 = 0;
  s = s - 1; } }

procedure Show {
 pink= difference;
 green = pink+ 1;
 blue = green + pink; }

procedure Enlarge {
 if pixel then {
  while dot {
    while notmove {
      line = edge + depth;
      semi = edge + increase - temporary + depth;
      call Fill;
      call Fill;
      edge = dot + 1 - decrease * temporary;
      if edge then {
```

```
        edge = 1 + (8 - temporary); }
       else {
        temporary = edge; }
       call Show;
       semi = temporary + edge;
       depth = semi * pixel + 1 - 3 * temporary;
       if notmove then {
        call Fill; }
       else {
        call Fill; }
       notmove = semi * half; }
     while dot {
      call Fill; }
     pixel = temporary * temporary; } }
  else {
   if pixel then {
    total = pixel * 1000; }
   else {
    while notdone {
     total = pixel + notdone; } } } }

procedure Fill {
  if temporary then {
   depth = depth + 1; }
  else {
   semi = depth - 1; } }

procedure Shrink {
  factor = incre - decrement;
  x1 = x1 - 10;
  x2 = x2 - 10;
```

```
  y1 = y1 - (10 * factor);
  y2 = y2 - (20 * factor);
  factor = y2 - y1 + x2 - x1;
  if factor then {
    while I {
      x1 = x1 - I;
      I = I - 1; }
    x2 = I * x1 - factor; }
  else {
    y2 = j * factor + incre;
    while j {
      j = j - 1;
      y1 = j * factor - decrement; } }
  call Draw;
  factor = factor * 0; }

procedure Translate {
  factor = 0;
  call Rotate; }

procedure Rotate {
  triangle = half * base * height;
  while edge {
    while line {
      if edge then {
        if pixel then {
          semi = temporary - depth + triangle; }
        else {
          dot = dot + degrees; }
        dx = dx + dy - triangle;
        base = dx - dy + dx - dy;
```

```
      height = base * dx * dy;
      edge = height + line * 2; }
    else {
      call Random; }
    dx = edge + triangle;
    triangle = triange + edge + dx; }
  call Show; } }

procedure Scale {
 if wrong then {
  while wcounter {
    location = unknown - wcounter; } }
 else {
  while wcounter {
    location = correct - wcounter; } } }

procedure PP {
 cs1 = 1;
 cs2 = 2;
 cs3 = 3;
 call QQ;
 call TT; }

procedure QQ {
 cs1 = cs2 * cs3; }

procedure RR {
 while cs4 {
  cs5 = 0;
  if cs1 then {
    call QQ; }
```

```
    else {
      call PP; } } }

procedure SS {
  call XX; }

procedure TT {
  call QQ;
  call UU;
  call SS; }

procedure UU {
  cs5 = 2;
  cs6 = 3;
  cs9 = 5;
  while cs9 {
    cs5 = cs5 - 1;
    if cs5 then {
      cs6 = cs5 + 1; }
    else {
      cs8 = cs6 + cs5; }
    cs6 = cs6 + (cs5 + cs9);
    call XX;
    cs9 = cs6 - 1; } }

procedure XX {
  if cs5 then {
    cs6 = cs5 + 1; }
  else {
    cs5 = cs6 + cs5; } }
```