Robotics

PS09 – Solutions

Andreas Birk

Constructor University

Part 9: Mapping

Problem 1

Given a quadtree with maximum depth 3, which is used to represent an area of 8×8 meters (or more precisely $[0.0,8.0[\times [0.0,8.0[$. Draw the quadtree after each sensor reading for the following sequence of sensor readings that indicate occupancy in the related locations:

1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

Quadtree Algorithm

quadrants: $q_x q_y$ = 00, 01, 10, 11

	qx	
qy	00	10
	01	11

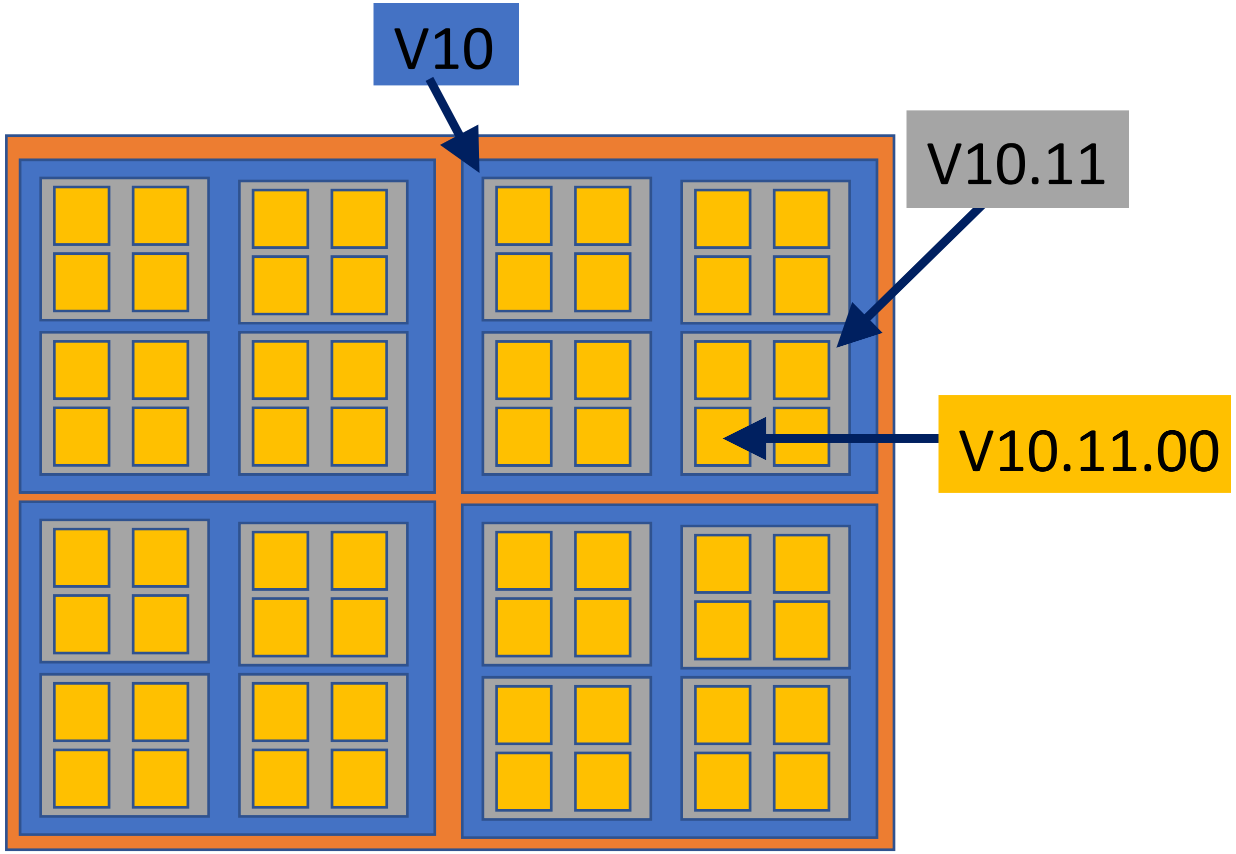
vertex names: vA.B

- A =parent index
- B = quadrant

depth

- 0 : root r
- 1 : v00 – v11
- 2 : vx.00 – vx.11
- 3 : vx.y.00 – vx.y.11

…



Quadtree Algorithm

quadtree $T = (V, E, L)$, maximum depth $d_{\max}$, point $(x, y) \in [0, x_{max}] \times [0, y_{max}]$
global variables: $v = root$, $x_m = x_{max}$, $y_m = y_{max}$, $d = 0$

quad-add((x, y))
  if $L(v) = \text{full}$: return
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor, q_y = \lfloor 2y/y_m \rfloor$
  if $\nexists\ v.q_x q_y$
    $V = V \cup \{v.q_x q_y\}, E = E \cup \{(v, v.q_x q_y)\}$
    if $d = d_{\max}$: $L(v.q_x q_y) = \text{full}$
  if $d = d_{\max}$
    test-full(v)
  else
    $v = v.q_x q_y$
    $x_m = x_m/2$, $y_m = y_m/2$
    $x = x - q_x \cdot x_m, y = y - q_y \cdot y_m$
    quad-add((x, y))

test-full(v)
  if $L(v.00) = L(v.01) = L(v.10) = L(v.11) = \text{full}$
    $V = V \backslash \{v.00, v.01, v.10, v.11\}$
    $E = E \backslash \{(v, v.00), (v, v.01), (v, v.10), (v, v.11)\}$
    $L(v) = \text{full}$
    test-full($\text{parent}(v)$)
  else
    return

Problem 1

1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)

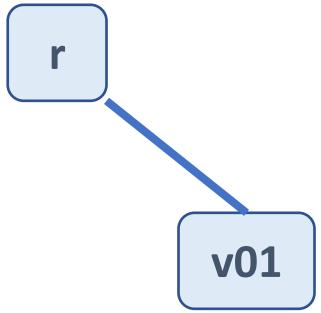
$(0.2, 7.1)$ $d = 0$: $v = r\,, x_m = 8\,, y_m = 8$

quad-add$((x, y))$
  if $L(v) =$ full : return
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor,\ q_y = \lfloor 2y/y_m \rfloor$ ⇐ $q_x = \left\lfloor 2 \cdot \frac{0.2}{8} \right\rfloor = 0\,,\ q_y = \left\lfloor 2 \cdot \frac{7.1}{8} \right\rfloor = 1$
  if $\nexists\ v.q_x q_y$
    $V = V \cup \{v.q_x q_y\},\ E = E \cup \{(v, v.q_x q_y)\}$
    if $d = d_{\max}$: $L(v.q_x q_y) = \text{full}$
  if $d = d_{\max}$
    test-full(v)
  else
    $v = v.q_x q_y$
    $x_m = x_m/2\,,\ y_m = y_m/2$
    $x = x - q_x \cdot x_m,\ y = y - q_y \cdot y_m$
    quad-add$((x, y))$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)



quad-add$((x, y))$

  if $L(v) = \text{full}$: return

  $d = d + 1$

  $q_x = \lfloor 2x/x_m \rfloor, q_y = \lfloor 2y/y_m \rfloor$ $q_x = 0\ , q_y = 1$

  if $\nexists\ v.q_x q_y$

    $V = V \cup \{v.q_x q_y\}, E = E \cup \{(v, v.q_x q_y)\}$ $v01$

    if $d = d_{\max}$: $L(v.q_x q_y) = \text{full}$

  if $d = d_{\max}$

    test-full(v)

  else

    $v = v.q_x q_y$

    $x_m = x_m/2\ , y_m = y_m/2$

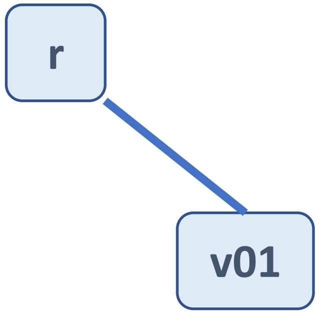
    $x = x - q_x \cdot x_m, y = y - q_y \cdot y_m$

    quad-add$((x, y))$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)



$d = 1\colon\ v = v01\,, x_m = 4\,, y_m = 4$

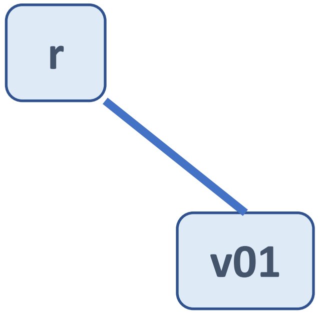
quad-add$((x, y))$
  if $L(v) = \text{full}$: return
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor,\ q_y = \lfloor 2y/y_m \rfloor$
  if $\nexists\ v.q_xq_y$
    $V = V \cup \{v.q_xq_y\},\ E = E \cup \{(v, v.q_xq_y)\}$
    if $d = d_{\max}\colon L(v.q_xq_y) = \text{full}$
  if $d = d_{\max}$
    test-full(v)
  else
    $v = v.q_xq_y$
    $x_m = x_m/2\,,\ y_m = y_m/2$
    $x = x - q_x \cdot x_m, y = y - q_y \cdot y_m$
    quad-add$((x, y))$ ⟵ $(0.2, 3.1)$

$x = 0.2 - 0 \cdot 4 = 0.2$
$y = 7.1 - 1 \cdot 4 = 3.1$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)



$d = 1\text{: } v = v01\,, x_m = 4\,, y_m = 4$

$(0.2, 3.1)$

quad-add$((x, y))$

if $L(v) = \text{full}$: return

$d = d + 1$

$q_x = \lfloor 2x/x_m \rfloor,\ q_y = \lfloor 2y/y_m \rfloor$ $\quad q_x = \left\lfloor 2 \cdot \frac{0.2}{4} \right\rfloor = 0\,,\ q_y = \left\lfloor 2 \cdot \frac{3.1}{4} \right\rfloor = 1$

if $\nexists\ v.q_x q_y$

$V = V \cup \{v.q_x q_y\},\ E = E \cup \{(v, v.q_x q_y)\}$

if $d = d_{\max}$: $L(v.q_x q_y) = \text{full}$

if $d = d_{\max}$

test-full(v)

else

$v = v.q_x q_y$

$x_m = x_m/2\,,\ y_m = y_m/2$

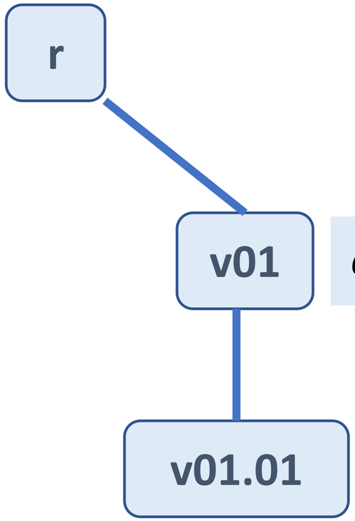
$x = x - q_x \cdot x_m,\ y = y - q_y \cdot y_m$

quad-add$((x, y))$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)



$d = 1\!: \; v = v01 \,, x_m = 4 \,, y_m = 4$

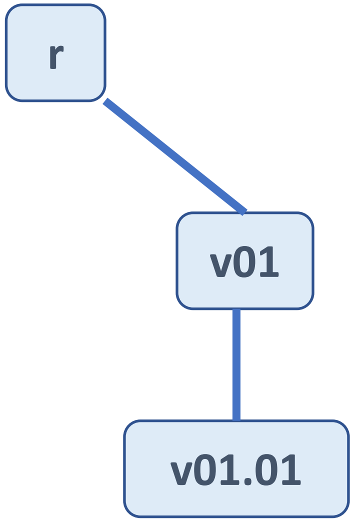
$(0.2, 3.1)$

quad-add$((x, y))$
  if $L(v) = \text{full}$: return
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor, \; q_y = \lfloor 2y/y_m \rfloor$ $q_x = 0, \; q_y = 1$
  if $\nexists \; v.q_x q_y$
    $V = V \cup \{v.q_x q_y\}, \; E = E \cup \{(v, v.q_x q_y)\}$ $v01.01$
    if $d = d_{\max}$: $L(v.q_x q_y) = \text{full}$
  if $d = d_{\max}$
    test-full(v)
  else
    $v = v.q_x q_y$
    $x_m = x_m/2 \,, y_m = y_m/2$
    $x = x - q_x \cdot x_m, \; y = y - q_y \cdot y_m$
    quad-add$((x, y))$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)



$d = 2$: $v = v01.01$, $x_m = 2$, $y_m = 2$

quad-add$((x, y))$
  if $L(v) = \text{full}$: return
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor$, $q_y = \lfloor 2y/y_m \rfloor$
  if $\nexists\, v.q_xq_y$
    $V = V \cup \{v.q_xq_y\}$, $E = E \cup \{(v, v.q_xq_y)\}$
    if $d = d_{\max}$: $L(v.q_xq_y) = \text{full}$
  if $d = d_{\max}$
    test-full(v)
  else
    $v = v.q_xq_y$
    $x_m = x_m/2$, $y_m = y_m/2$
    $x = x - q_x \cdot x_m$, $y = y - q_y \cdot y_m$
    quad-add$((x, y))$

$(0.2, 1.1)$

$x = 0.2 - 0 \cdot 2 = 0.2$
$y = 3.1 - 1 \cdot 2 = 1.1$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

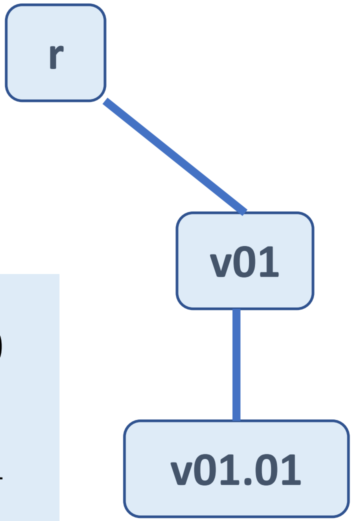
Problem 1

1. (0.2, 7.1)

$(0.2, 1.1)$

quad-add$((x, y))$
  if $L(v) = \text{full}$: return
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor, q_y = \lfloor 2y/y_m \rfloor$
  if $\nexists\ v.q_xq_y$
    $V = V \cup \{v.q_xq_y\}, E = E \cup \{(v, v.q_xq_y)\}$
    if $d = d_{\max}$: $L(v.q_xq_y) = \text{full}$
  if $d = d_{\max}$
    test-full(v)
  else
    $v = v.q_xq_y$
    $x_m = x_m/2 , y_m = y_m/2$
    $x = x - q_x \cdot x_m, y = y - q_y \cdot y_m$
    quad-add$((x, y))$

$$q_x = \left\lfloor 2 \cdot \frac{0.2}{2} \right\rfloor = 0$$
$$q_y = \left\lfloor 2 \cdot \frac{1.1}{2} \right\rfloor = 1$$

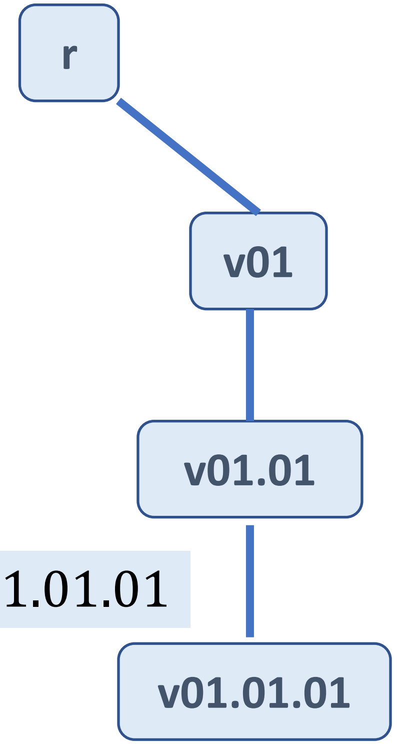


$d = 2$: $v = v01.01 , x_m = 2 , y_m = 2$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)



$(0.2, 1.1)$

quad-add$((x, y))$

  if $L(v) = \text{full}$: return

  $d = d + 1$

  $q_x = \lfloor 2x/x_m \rfloor,\ q_y = \lfloor 2y/y_m \rfloor$ $\quad q_x = 0,\ q_y = 1$

  if $\nexists\ v.q_xq_y$

    $V = V \cup \{v.q_xq_y\},\ E = E \cup \{(v, v.q_xq_y)\}$ $\quad v01.01.01$

    if $d = d_{\max}$: $L(v.q_xq_y) = \text{full}$

  if $d = d_{\max}$

    test-full(v)

  else

    $v = v.q_xq_y$

    $x_m = x_m/2$, $y_m = y_m/2$

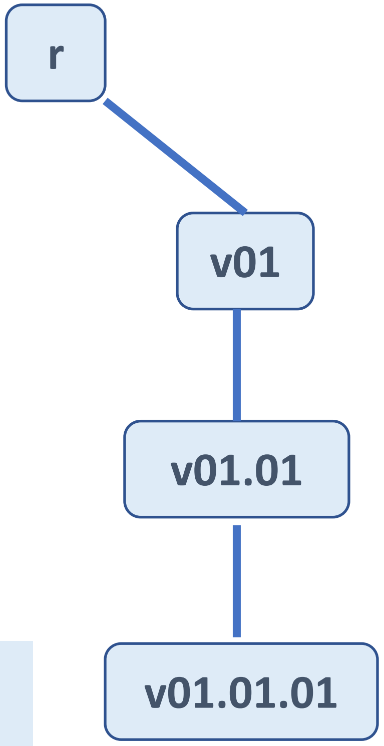
    $x = x - q_x \cdot x_m,\ y = y - q_y \cdot y_m$

    quad-add$((x, y))$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

1. (0.2, 7.1)



$d = 3$: $L(v01.01.01) = \text{full}$

quad-add$((x, y))$
  if $L(v) = \text{full}$: return
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor, q_y = \lfloor 2y/y_m \rfloor$
  if $\nexists\, v.q_x q_y$
    $V = V \cup \{v.q_x q_y\}, E = E \cup \{(v, v.q_x q_y)\}$
    if $d = d_{\max}$: $L(v.q_x q_y) = \text{full}$

$d = 3 = d_{max}$

  if $d = d_{\max}$
    test-full(v)

$v = v01.01$

  else
    $v = v.q_x q_y$
    $x_m = x_m/2$, $y_m = y_m/2$
    $x = x - q_x \cdot x_m, y = y - q_y \cdot y_m$
    quad-add$((x, y))$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

$v = v01.01$

test-full(v)

  if $L(v.00) = L(v.01) = L(v.10) = L(v.11) = \text{full}$ false

    $V = V \backslash \{v.00, v.01, v.10, v.11\}$

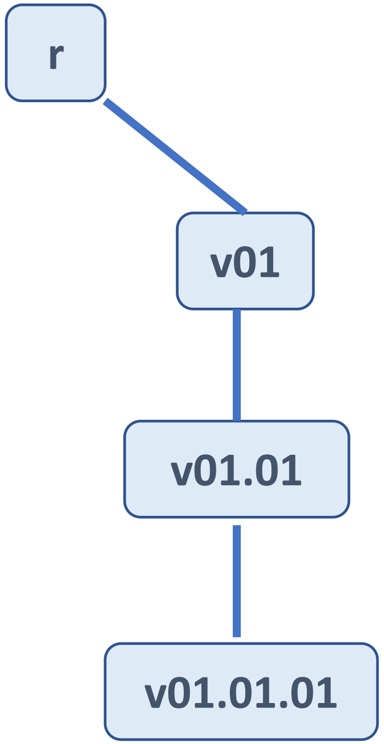
    $E = E \backslash \{(v, v.00), (v, v.01), (v, v.10), (v, v.11)\}$

    $L(v) = \text{full}$

    test-full($\text{parent}(v)$)

  else

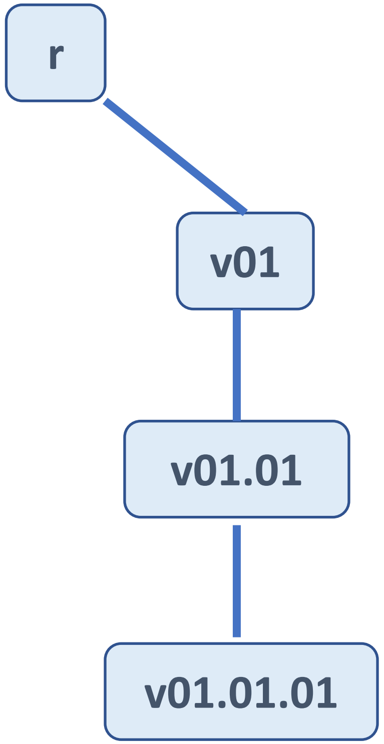
    return



1. (0.2, 7.1)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

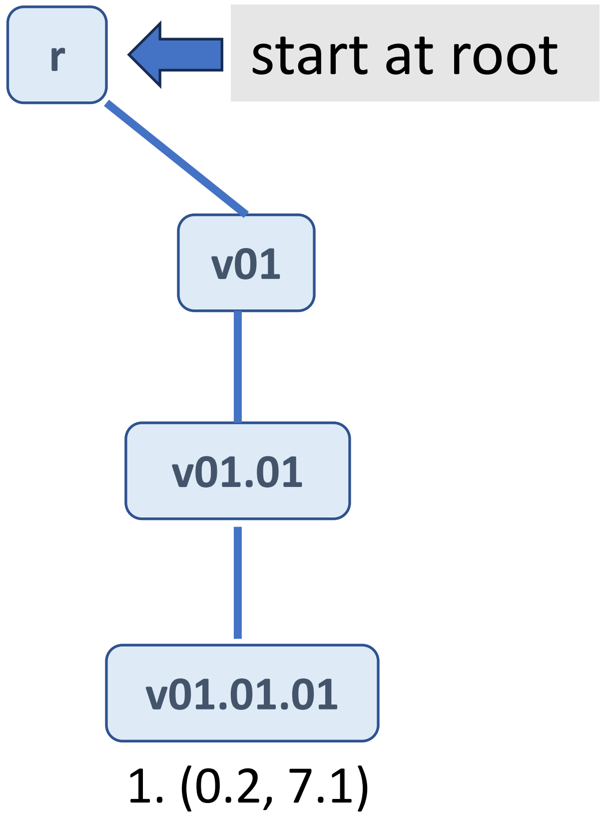


1. (0.2, 7.1)

1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

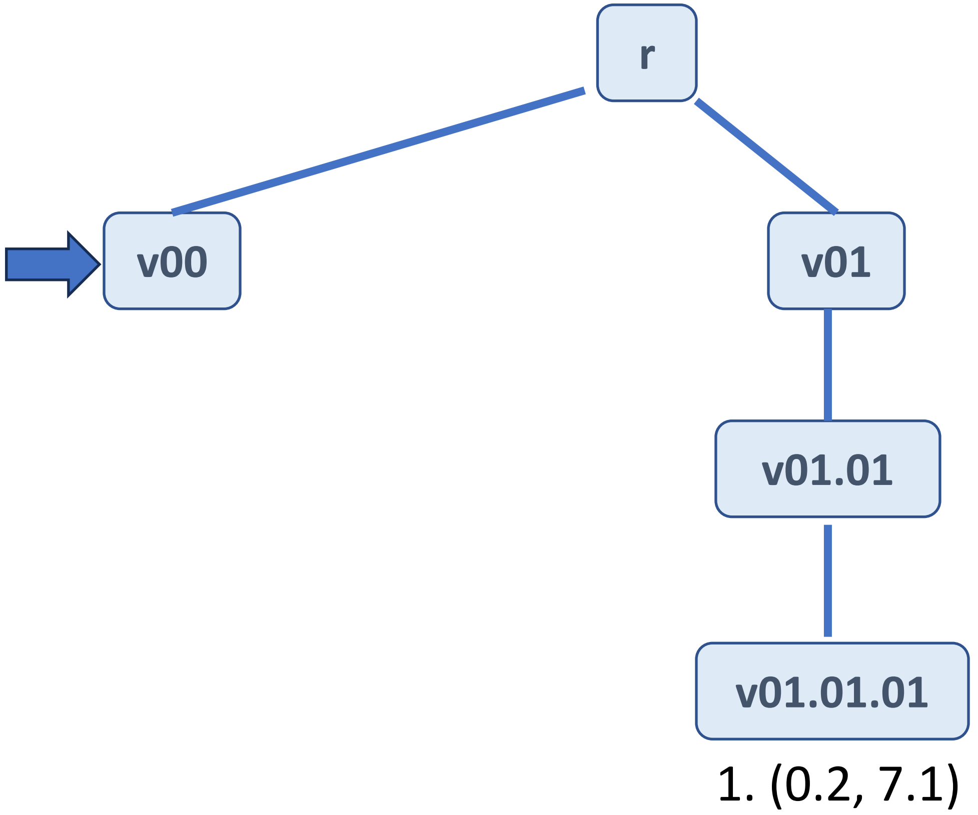


1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

new vertex,
continue recursion

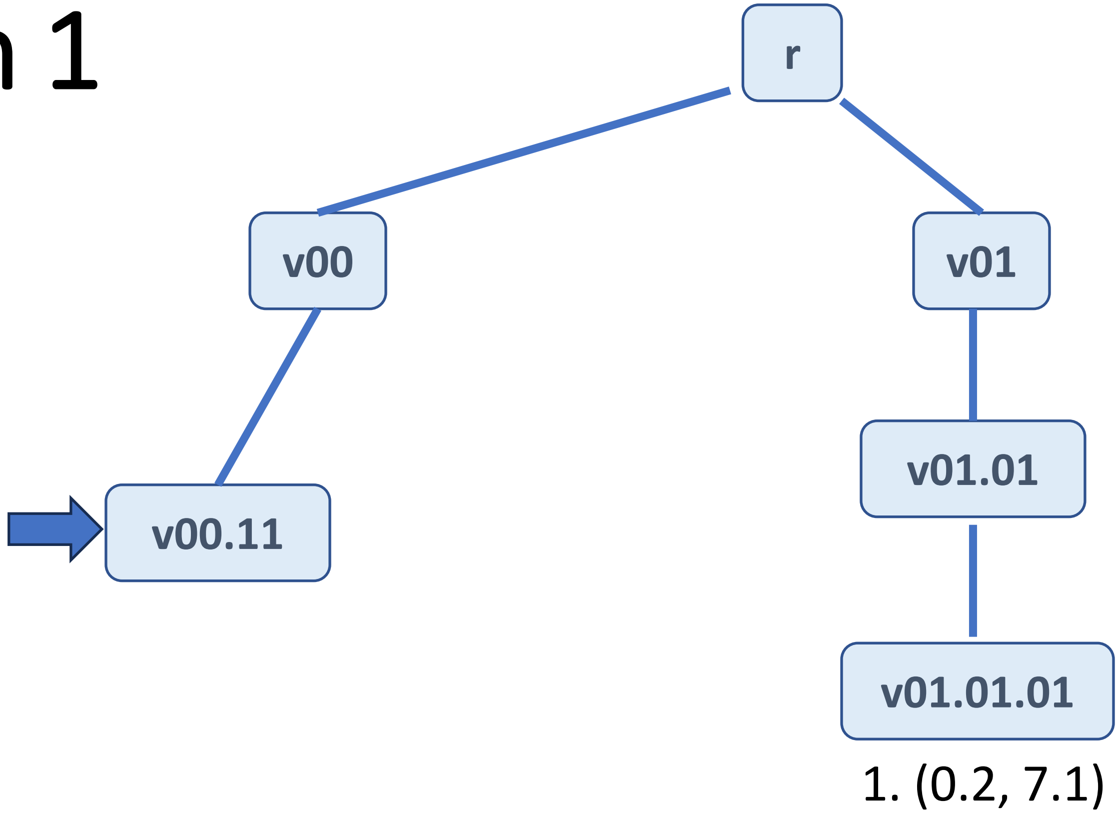


1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

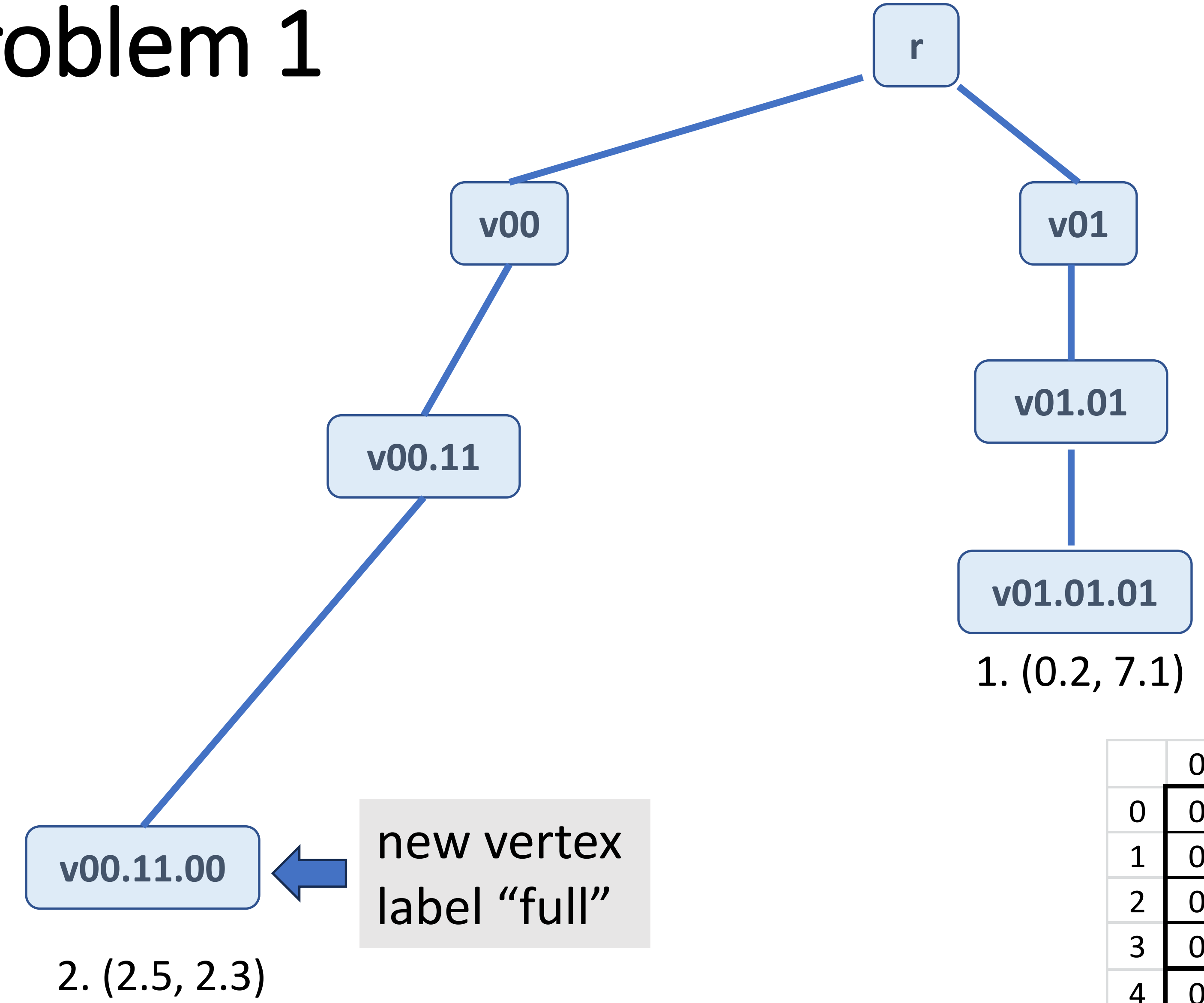
new vertex,
continue recursion



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

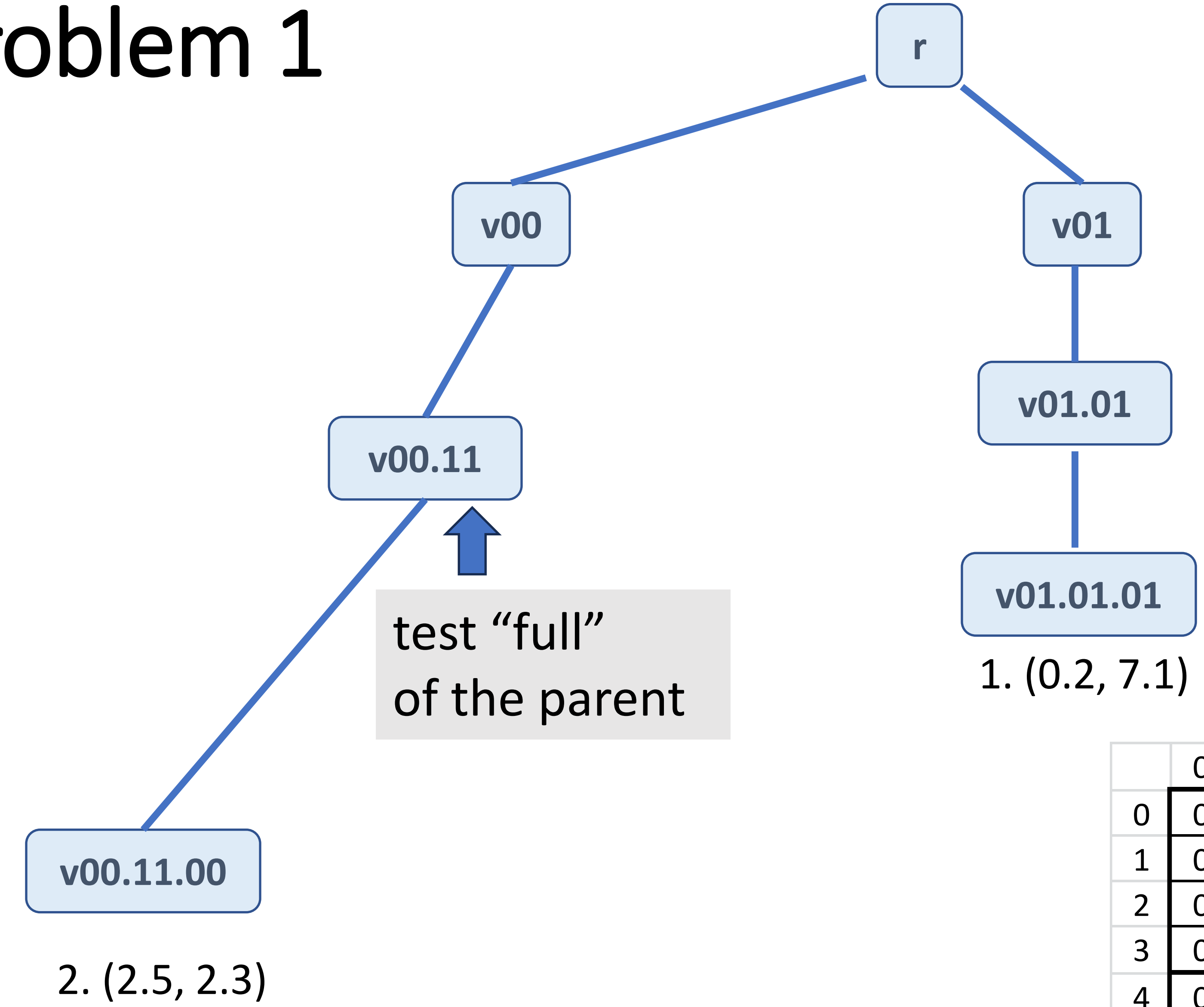
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

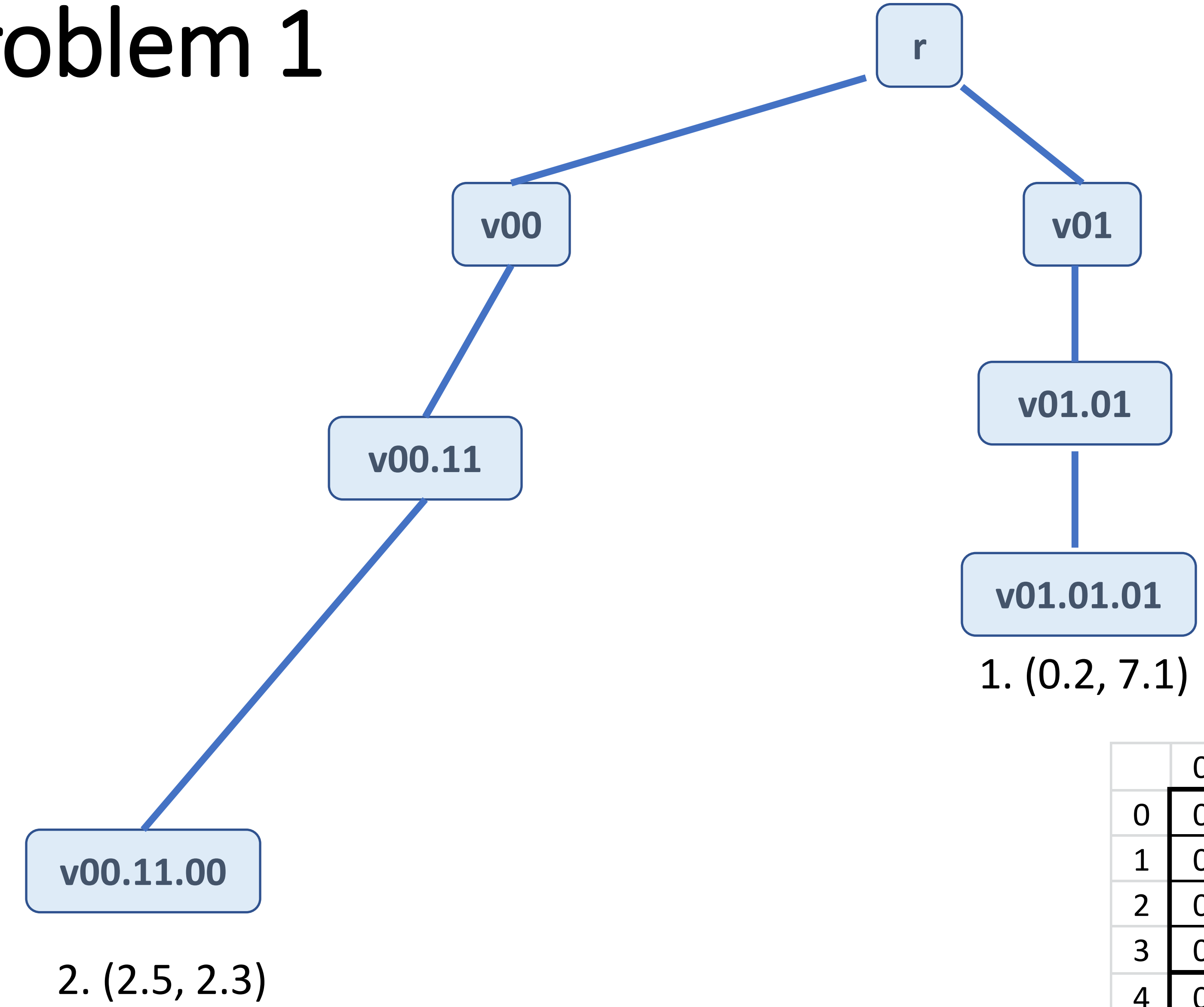
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

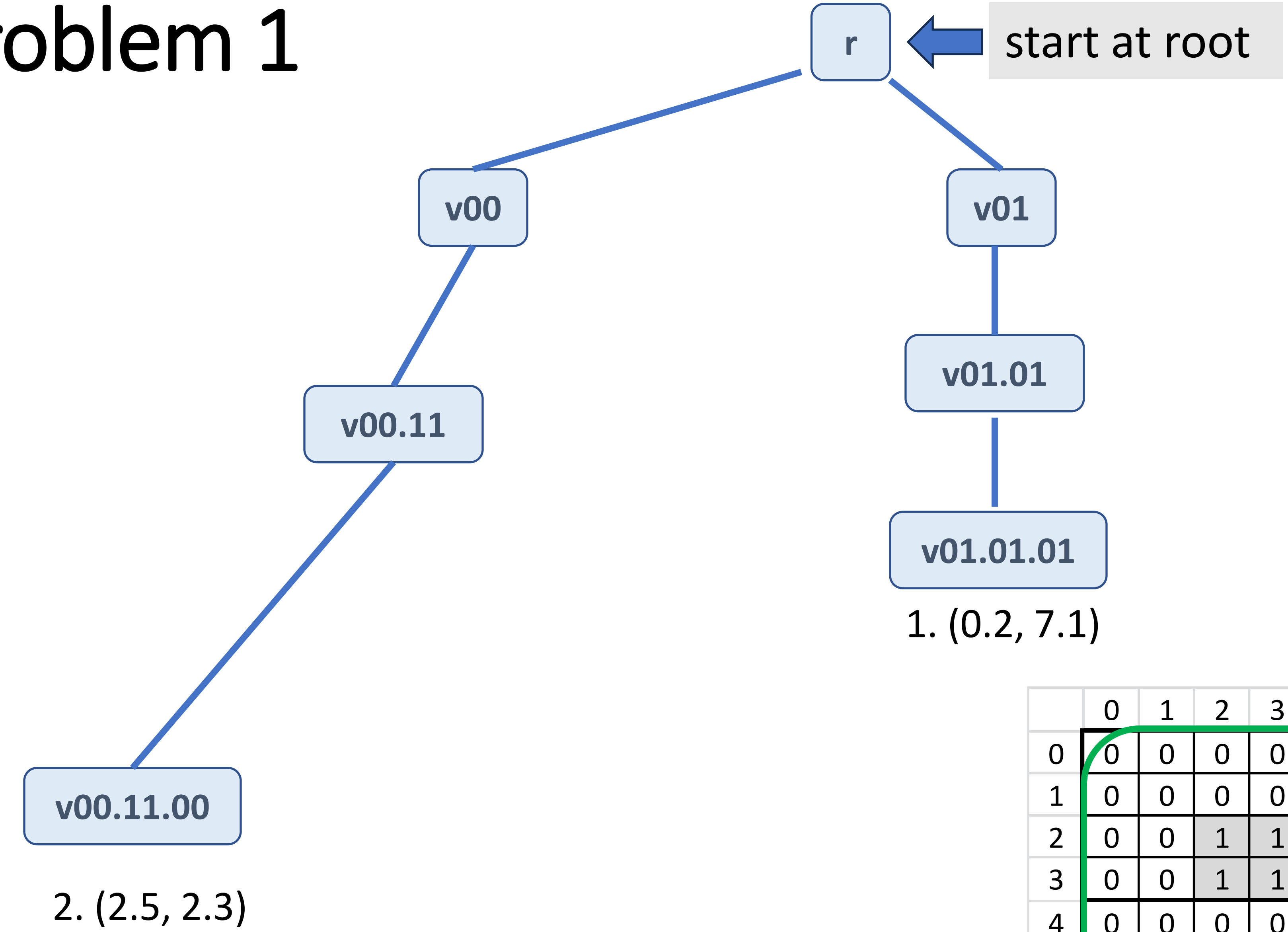
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

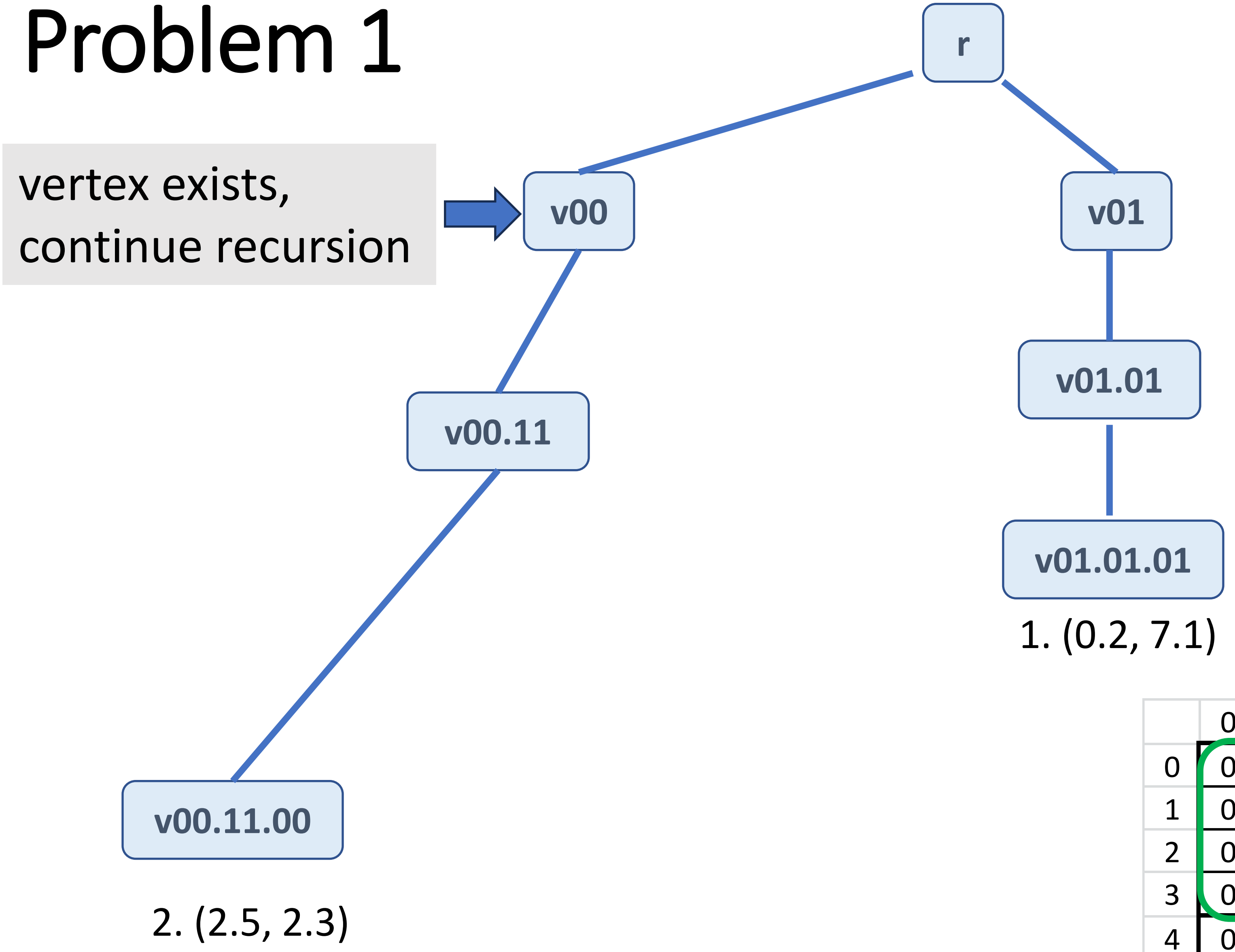
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

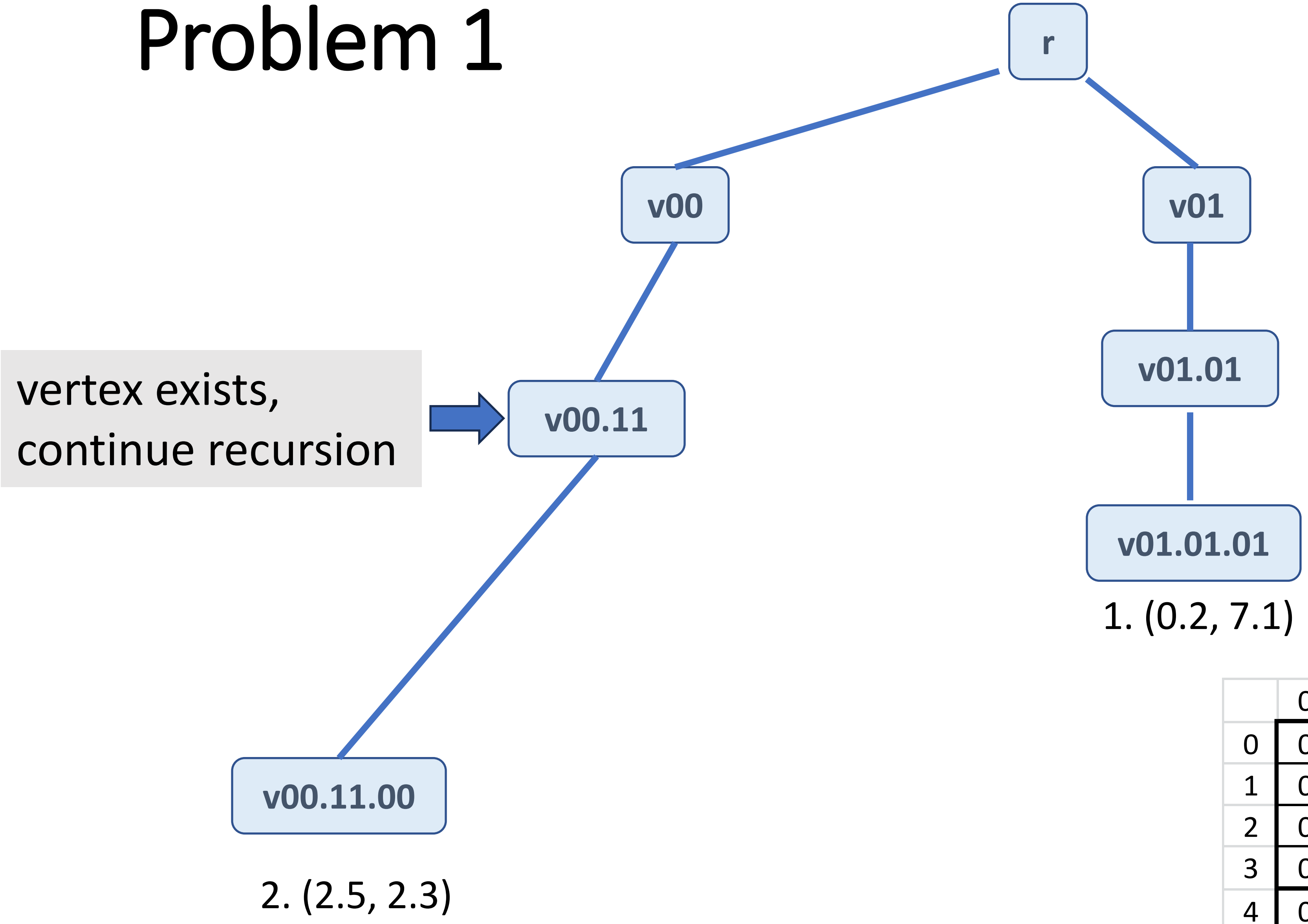
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

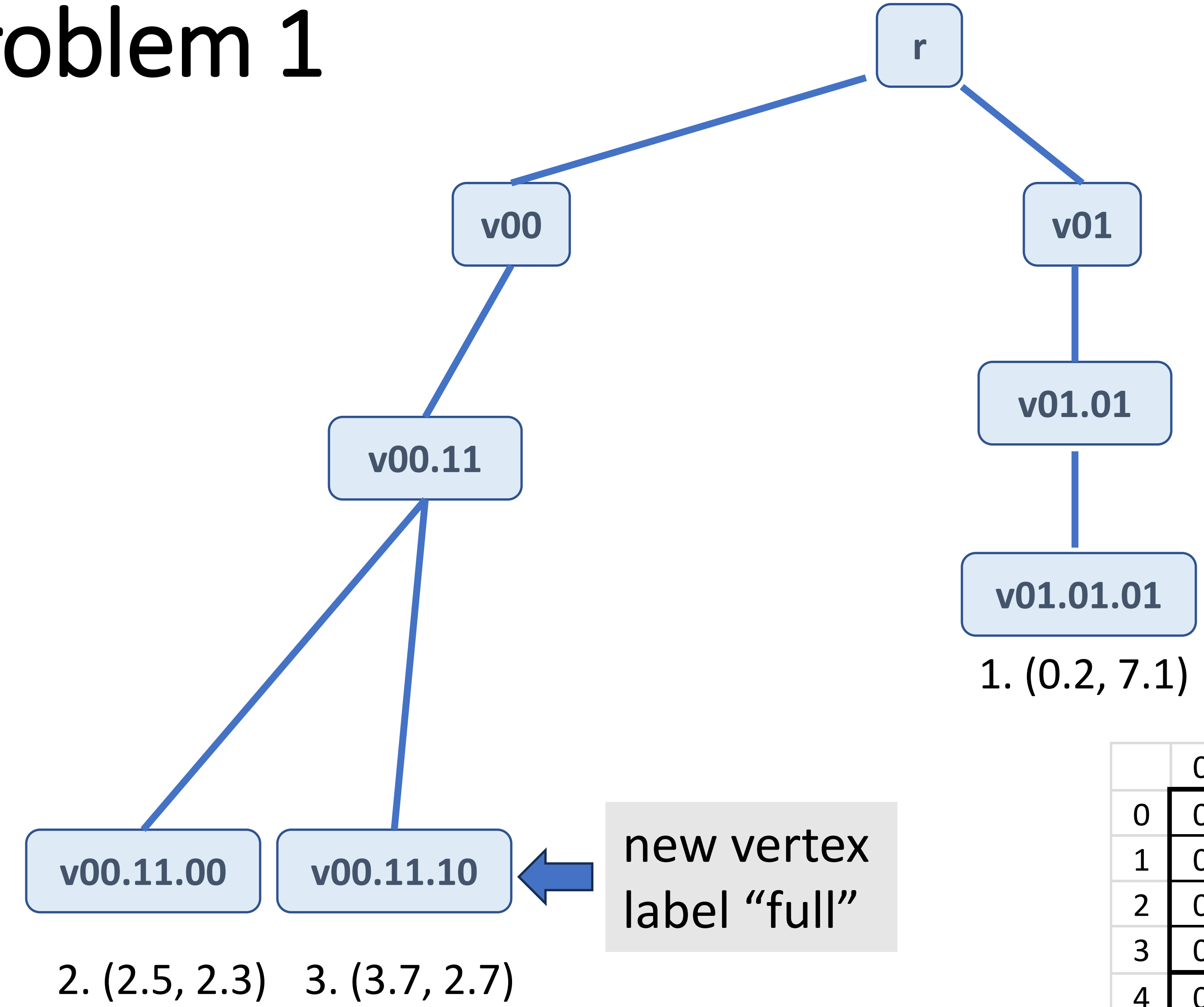
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

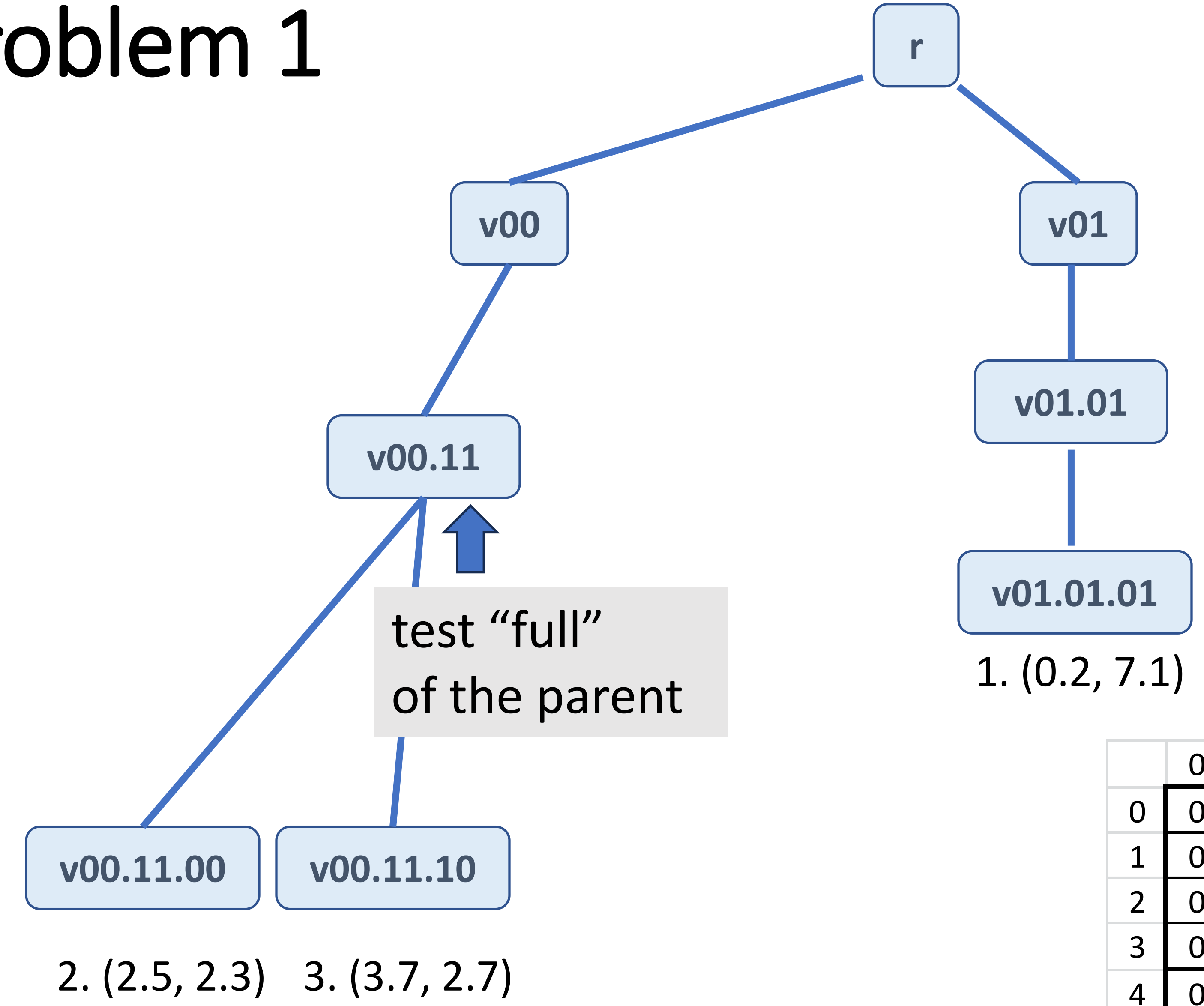
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

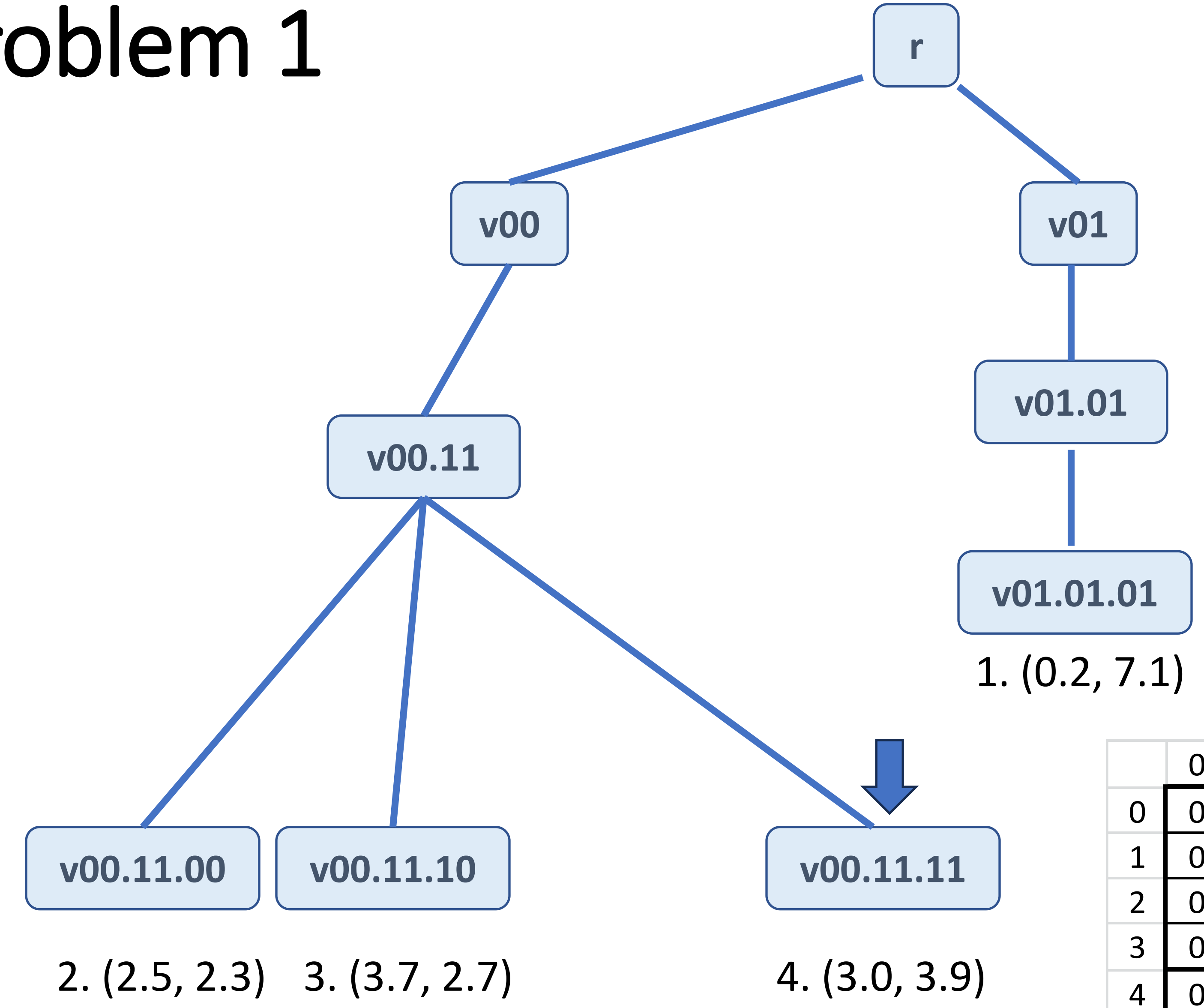
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

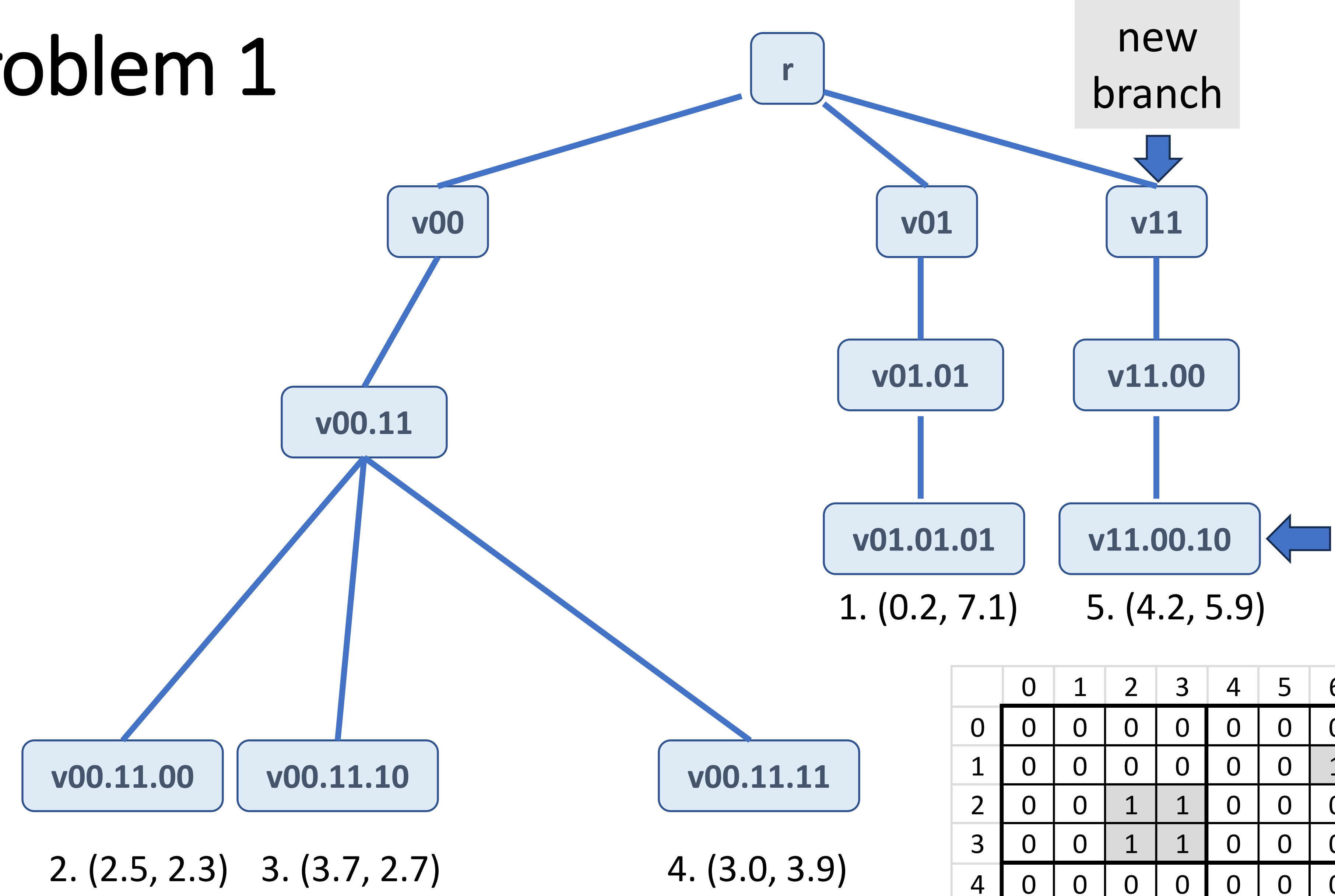


similar to previous

1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

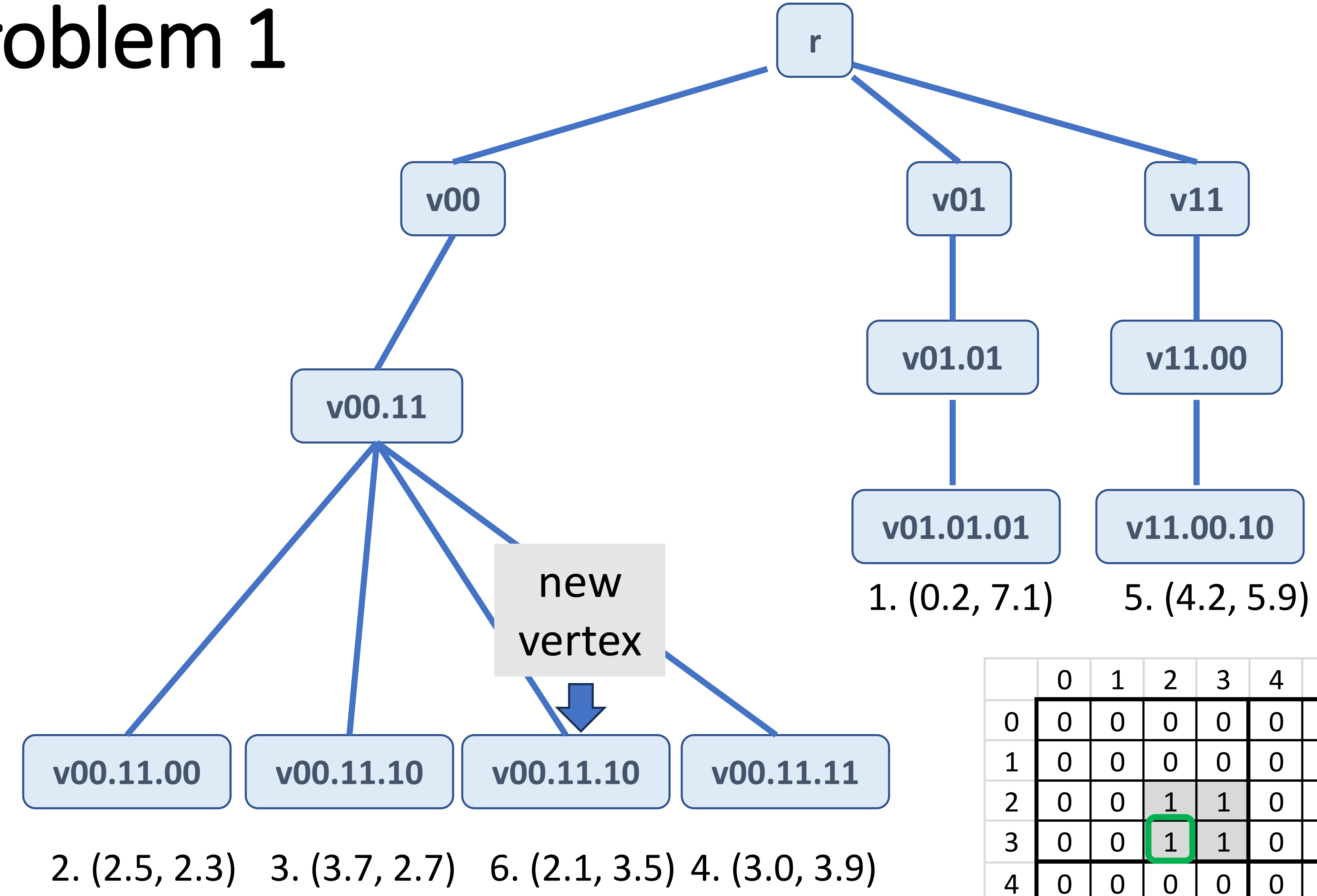
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

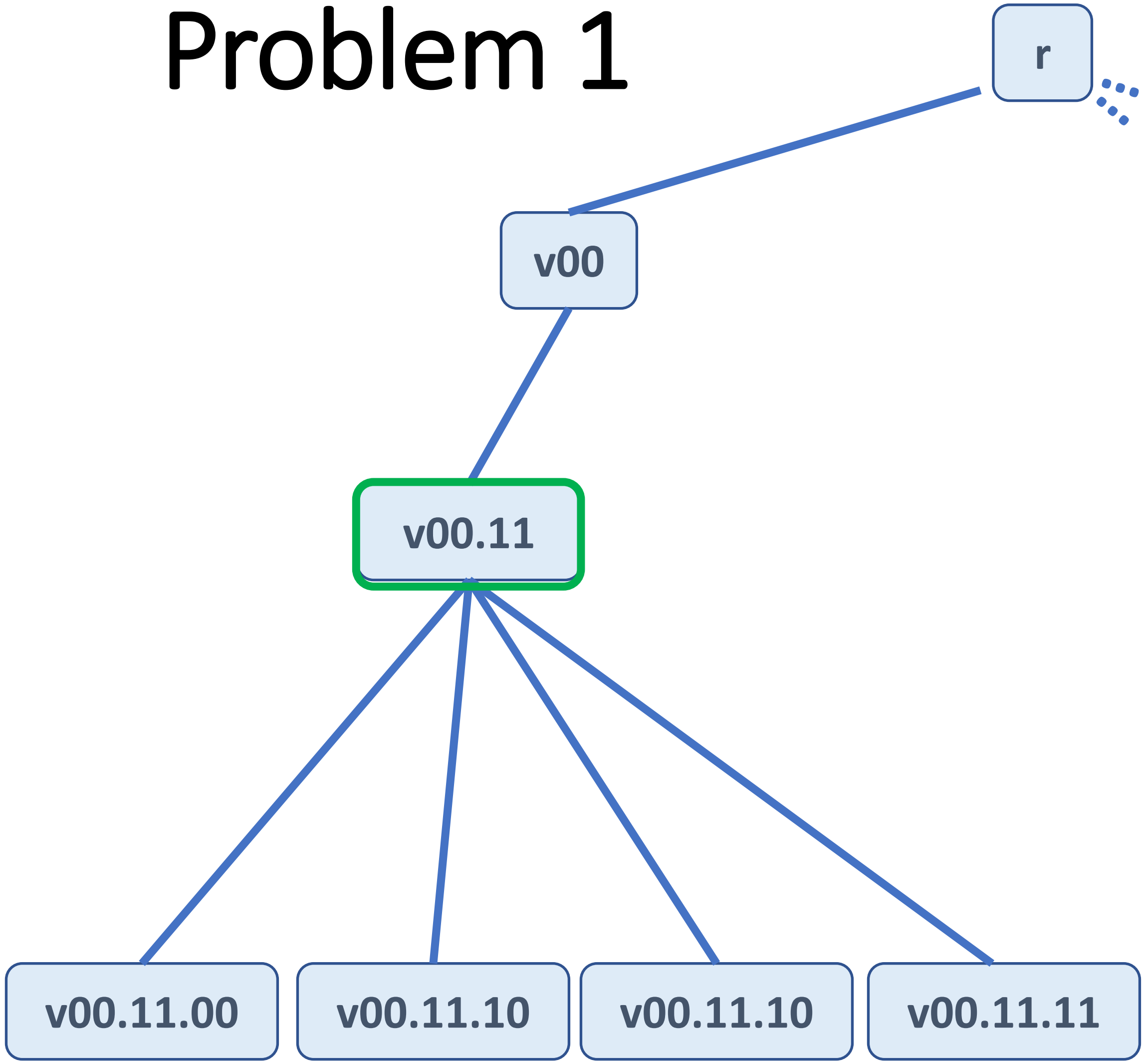
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1



2. (2.5, 2.3)   3. (3.7, 2.7)   6. (2.1, 3.5)   4. (3.0, 3.9)

test-full(v) $v = v10.11$

if $L(v.00) = L(v.01) = L(v.10) = L(v.11) = \text{full}$ true

$V = V \setminus \{v.00, v.01, v.10, v.11\}$

$E = E \setminus \{(v, v.00), (v, v.01), (v, v.10), (v, v.11)\}$

$L(v) = \text{full}$

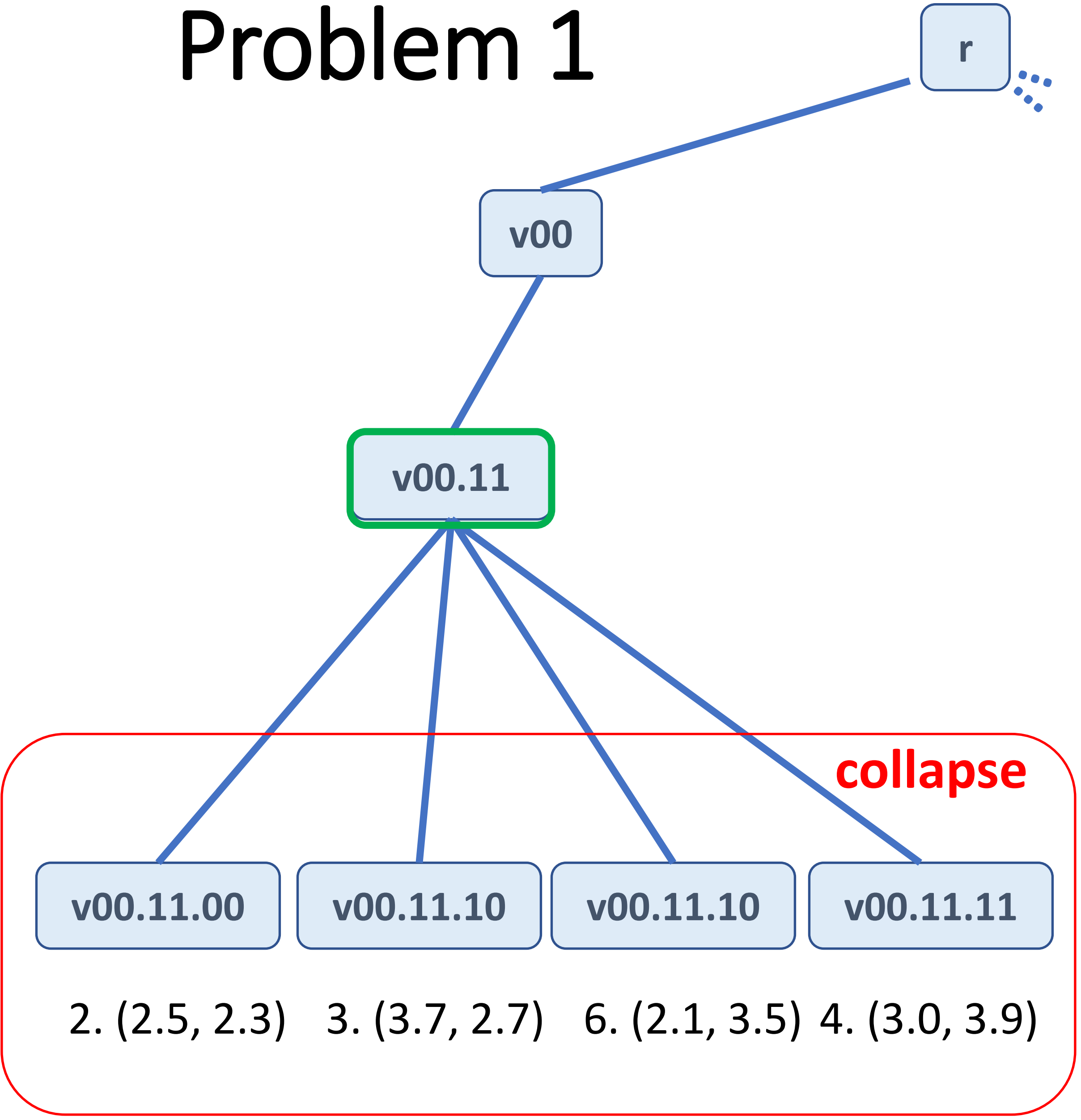
test-full$(\text{parent}(v))$

else

return

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1



test-full(v) $v = v10.11$

if $L(v.00) = L(v.01) = L(v.10) = L(v.11) = \text{full}$ true

$V = V \setminus \{v.00, v.01, v.10, v.11\}$

$E = E \setminus \{(v, v.00), (v, v.01), (v, v.10), (v, v.11)\}$

$L(v) = \text{full}$

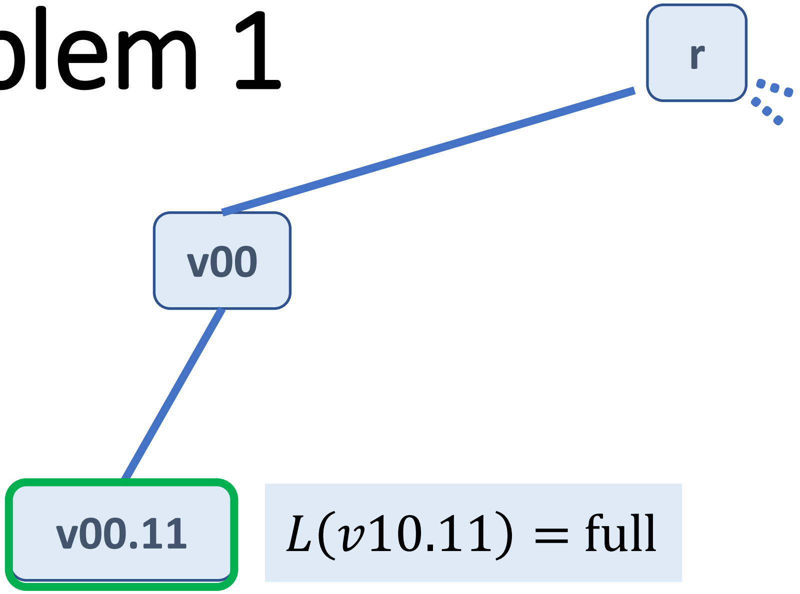
test-full(parent(v))

else

return

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1



$L(v10.11) = \text{full}$

2. (2.5, 2.3)

3. (3.7, 2.7)

4. (3.0, 3.9)

6. (2.1, 3.5)

test-full(v) $v = v10.11$

if $L(v.00) = L(v.01) = L(v.10) = L(v.11) = \text{full}$ true

$V = V \backslash \{v.00, v.01, v.10, v.11\}$

$E = E \backslash \{(v, v.00), (v, v.01), (v, v.10), (v, v.11)\}$

$L(v) = \text{full}$

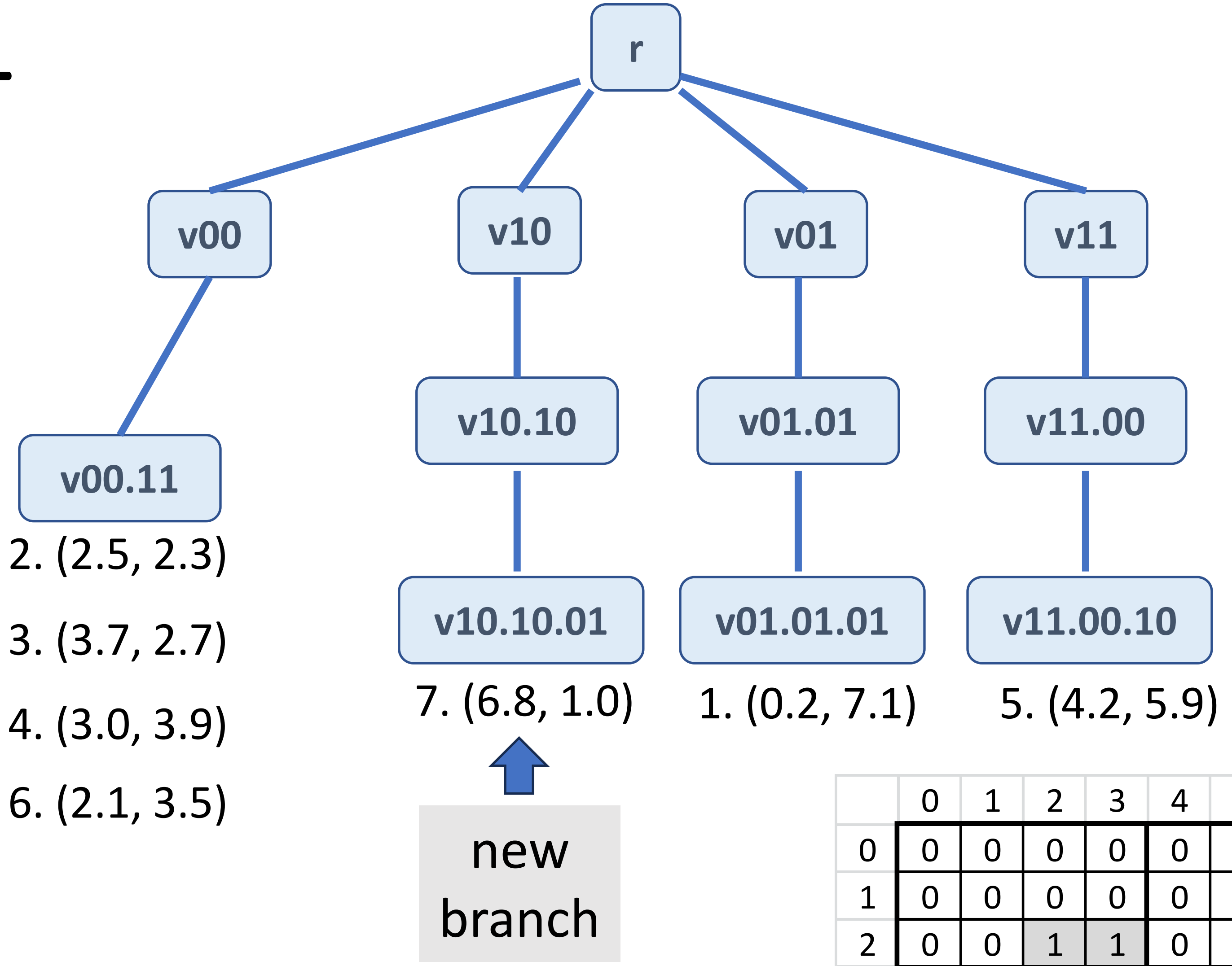
test-full$(\text{parent}(v))$

else

return

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

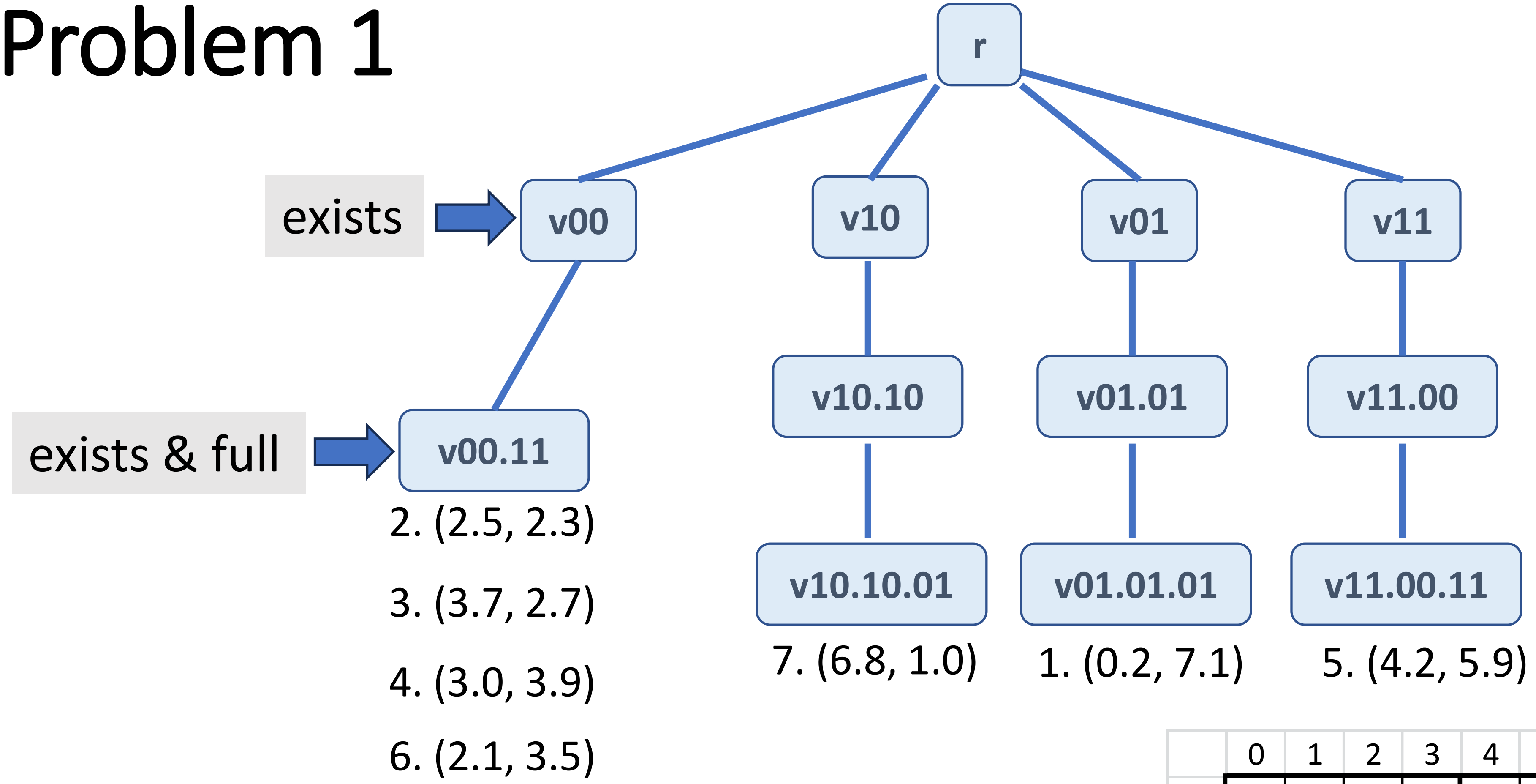
Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

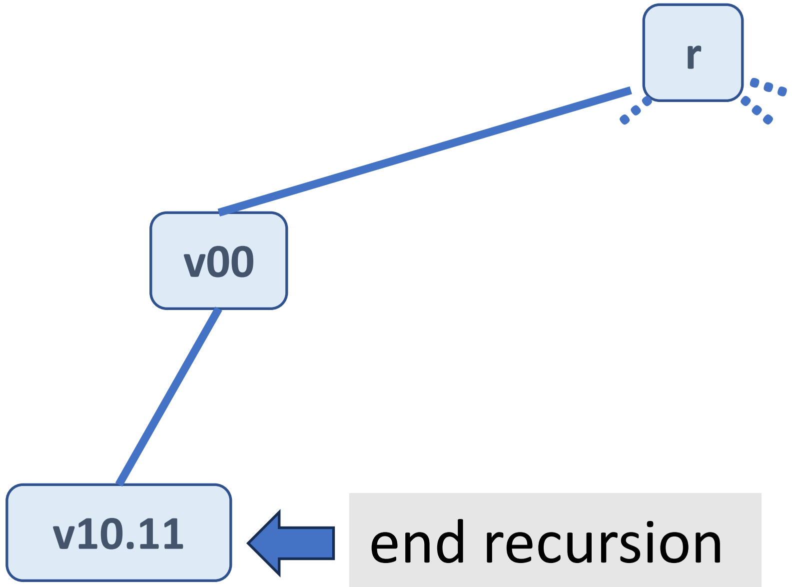


1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1

quad-add$((x, y))$
  if $L(v) = \text{full}$: return $L(v10.11) = \text{full}$
  $d = d + 1$
  $q_x = \lfloor 2x/x_m \rfloor$, $q_y = \lfloor 2y/y_m \rfloor$
  if $\nexists\ v.q_x q_y$
    $V = V \cup \{v.q_x q_y\}$, $E = E \cup \{(v, v.q_x q_y)\}$
    if $d = d_{\max}$: $L(v.q_x q_y) = \text{full}$
  if $d = d_{\max}$
    test-full(v)
  else
    $v = v.q_x q_y$
    $x_m = x_m/2$, $y_m = y_m/2$
    $x = x - q_x \cdot x_m$, $y = y - q_y \cdot y_m$
    quad-add$((x, y))$

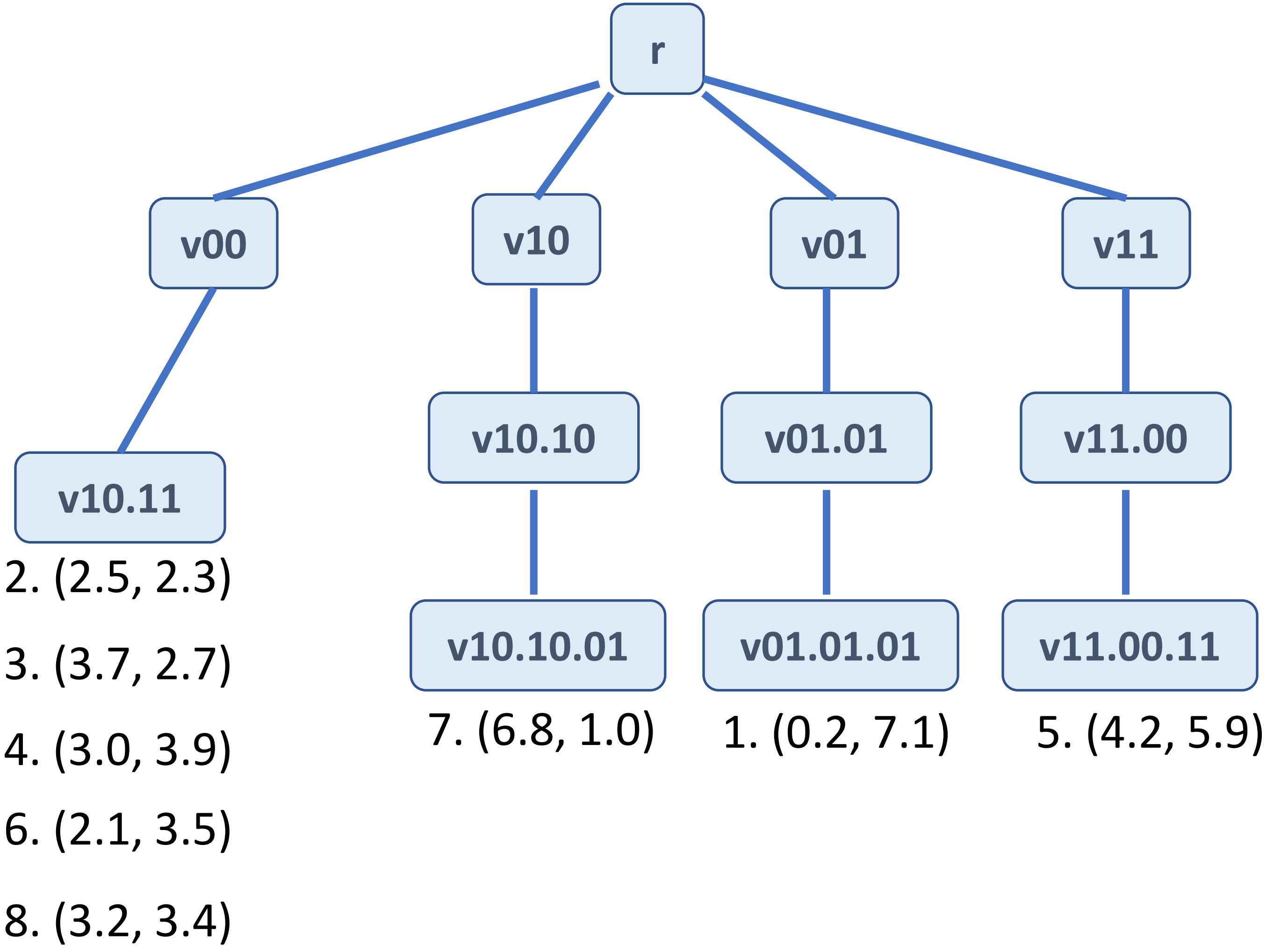


2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
6. (2.1, 3.5)

1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 1



1. (0.2, 7.1)
2. (2.5, 2.3)
3. (3.7, 2.7)
4. (3.0, 3.9)
5. (4.2, 5.9)
6. (2.1, 3.5)
7. (6.8, 1.0)
8. (3.2, 3.4)

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0

Problem 2

Consider a 1-dimensional world where a mobile robot r has a 1-dimensional range sensor that returns the distance d_o to the nearest obstacle. An evidence grid $g(x)$ with log odds is to be used for representing uncertainty in a map of the environment. Concretely, a base two logarithm ($\log_2$) is used for the log odds.

The robot is supposed to generate a 1D map over 5 cm with a 1 cm resolution, i.e., $g(x)$ holds the occupancy estimate of the area $[x\ cm, x+1\ cm[$. For the sake of convenience, we assume discrete motions and discrete sensor readings.

Given the robot pose x_r and a sensor reading d_o, the conditional probability $P(s = d_o|o@x)$, respectively $P(s = d_o|\neg o@x)$ - or short $P(o@x)$ and $P(\neg o@x)$ - of getting sensor value d_o when there is an obstacle at x ("$o@x$"), respectively free space at coordinate x ("$\neg o@x$") is given as:

- $P(o@x) = 0.9 \Leftrightarrow x = x_r + d_o$
- $P(o@x) = 0.3 \Leftrightarrow x = x_r + d_o \pm 1cm$
- $P(\neg o@x) = 0.8 \Leftrightarrow \forall x : x_r \leq x < x_r + d_o - 1cm$
- for all other x it holds that $P(o@x) = P(\neg o@x)$

No information about the environment is given for the initial state of the map, i.e., $\forall x : P(o@x) = P(\neg o@x)$ as long as there are no sensor readings yet.

- What is the initial map $g(.)_0$ at time $t = 0$, i.e., the value of all cells $g(x)_0$?
- Suppose the robot starts at coordinate (0) and gets a sensor reading of $d_o = 6$ at $t = 1$. What does the map $g(.)_1$ look like after this sensor reading is integrated in it?
- At $t = 2$, the robot is moving and it gets to coordinate (3). There, the sensor value is $d_o = 4$. What does the map $g(.)_2$ look like after this sensor reading is used to update the map?
- At $t = 3$, the robot is still at coordinate (3). The sensor value is now $d_o = 3$. What does the map $g(.)_3$ look like?
- At $t = 4$, the robot is again still at coordinate (3). The sensor value is again $d_o = 3$. What does the map $g(.)_4$ look like?

Problem 2

Consider a 1-dimensional world where a mobile robot r has a 1-dimensional range sensor that returns the distance d_o to the nearest obstacle. An evidence grid $g(x)$ with log odds is to be used for representing uncertainty in a map of the environment. Concretely, a base two logarithm ($\log_2$) is used for the log odds.

The robot is supposed to generate a 1D map over 10 cm with a 1 cm resolution, i.e., $g(x)$ holds the occupancy estimate of the area $[x\ cm, x + 1\ cm[$. For the sake of convenience, we assume discrete motions and discrete sensor readings.

x coordinate ->	0	1	2	3	4	5	6	7	8	9
map ->										

Problem 2

…
Given the robot pose x_r and a sensor reading d_o, the conditional probability $P(s = d_o | o@x)$, respectively $P(s = d_o | \neg o@x)$ - or short $P(o@x)$ and $P(\neg o@x)$ - of getting sensor value d_o when there is an obstacle at x ("$o@x$"), respectively free space at coordinate x ("$\neg o@x$") is given as:

- $P(o@x) = 0.9 \Leftrightarrow x = x_r + d_o$
- $P(o@x) = 0.3 \Leftrightarrow x = x_r + d_o \pm 1cm$
- $P(\neg o@x) = 0.8 \Leftrightarrow \forall x : x_r \leq x < x_r + d_o - 1cm$

…

Problem 2

- $P(o@x) = 0.9 \Leftrightarrow x = x_r + d_o$
- $P(o@x) = 0.3 \Leftrightarrow x = x_r + d_o \pm 1cm$
- $P(\neg o@x) = 0.8 \Leftrightarrow \forall x : x_r \leq x < x_r + d_o - 1cm$

log odds

- $\log_2\left(\frac{P(o@(x=x_r+d_o))}{P(\neg o@(x=x_r+d_o))}\right) = \log_2\left(\frac{0.9}{0.1}\right) = 3.169925$
- $\log_2\left(\frac{P(o@(x=x_r+d_o\pm 1))}{P(\neg o@(x=x_r+d_o\pm 1))}\right) = \log_2\left(\frac{0.3}{0.7}\right) = -1.22239$
- $\log_2\left(\frac{P(o@(x<x_r+d_o-1))}{P(\neg o@(x<x_r+d_o-1))}\right) = \log_2\left(\frac{0.2}{0.8}\right) = -2$

Problem 2

…

No information about the environment is given for the initial state of the map, i.e., $\forall x :$ $P(o@x) = P(\neg o@x)$ as long as there are no sensor readings yet.

- What is the initial map $g(.)_0$ at time $t = 0$, i.e., the value of all cells $g(x)_0$?

…

$$P(o@x) = P(\neg o@x) = 0.5 \Rightarrow \log_2\left(\frac{0.5}{0.5}\right) = 0$$

t =		0	1	2	3	4	5	6	7	8	9
0	initial map	0	0	0	0	0	0	0	0	0	0

Problem 2

…

- Suppose the robot starts at coordinate (0) and gets a sensor reading of $d_o = 6$ at $t = 1$. What does the map $g(.)_1$ look like after this sensor reading is integrated in it?
- At $t = 2$, the robot is moving and it gets to coordinate (3). There, the sensor value is $d_o = 4$. What does the map $g(.)_2$ look like after this sensor reading is used to update the map?
- At $t = 3$, the robot is still at coordinate (3). The sensor value is now $d_o = 3$. What does the map $g(.)_3$ look like?
- At $t = 4$, the robot is again still at coordinate (3). The sensor value is again $d_o = 3$. What does the map $g(.)_4$ look like?

Problem 2

$$\log_2\left(\frac{P(o@<d_o-1))}{P(\neg o@<d_o-1))}\right)=-2, \log_2\left(\frac{P(o@d_o\pm 1)}{P(\neg o@d_o\pm 1)}\right)=-1.22239, \log_2\left(\frac{P(o@d_o)}{P(\neg o@d_o)}\right)=3.169925$$

t =		0	1	2	3	4	5	6	7	8	9
0	initial map	0	0	0	0	0	0	0	0	0	0
1	robot@(0), d=6	-2	-2	-2	-2	-2	-1.22	3.17	-1.22	0	0
2	robot@(3), d=4	-2	-2	-2	-4	-4	-3.22	1.95	1.95	-1.22	0
3	robot@(3), d=3	-2	-2	-2	-6	-6	-4.44	5.12	0.73	-1.22	0
4	robot@(3), d=3	-2	-2	-2	-8	-8	-5.67	8.29	-0.5	-1.22	0