An Optimal Algorithm for Finding the Spectrum of the Joint Action of Binary Factors

Julia Nagrebetskaya, Vladimir Panov, Ekaterina Vasilko

Ural Federal University
Department of Mathematics, Mechanics and Computer Science
Institute of Industrial Ecology of the Ural Branch of RAS

Yekaterinburg, Russia

01.05.2023

Binary theory of sufficient causes is a causality model commonly used in biomedical sciences [Rothman (1976), Miettinen (1982), VanderWeele and Robins (2006)]. Firstly it was suggested in philosophy [Mackie, Lewis, 1965-1987] and simultaneously much in the same fashion in epidemiology [MacMahon and Pugh]. Binary model deals with two-level factors and response [Miettinen (1982), Greenland and Poole (1988), VanderWeele, Robins, Richardson (2006-2012)].

Mathematical formalization of the binary sufficient causes theory based on the Boolean framework. This formalization allows one to make significant progress in the formal description of that model, as well as to introduce a number of new concepts. In particular, this concerns the notion of the spectrum M_f of joint action of factors in a given response f and algorithms (Algorithm 1 and Algorithm 2) for its effective calculation [Authors (2019), USEBEREIT (2022)].

Algorithms 1 and 2 are implemented in Java and C#, respectively. Here we propose a modification of the Algorithm 1, refered to as Algorithm 3 below, which is based on the nonrecursive preoder traversal of the nodes of a tree. The Algorithm 3 is implemented in Java, similarly to the Algorithm 1.

Recall the notion of joint action spectrum M_f of binary factors in a binary response f and related concepts.

Definition 1 [Authors (2019)]

We call *degree of joint action of factors* $x = (x_1, ..., x_n)$ in a response $f \in \mathbb{B}(x_1, x_2, ..., x_n)$ at values of factors $x = \alpha$, where $\alpha \in C_f$, a number $\mu_f(\alpha)$ defined by equality $\mu_f(\alpha) = d(\alpha, C_f \backslash \{\alpha\}) - 1$, if $|C_f| > 1$, and $\mu_f(\alpha) = n$ if $|C_f| = 1$.

Here $|C_f|$ is the cardinality of the set $C_f = \{\alpha \in \mathbb{B}^n \mid f(\alpha) = 1\}$, $d(\alpha, \beta)$ is the Hamming distance between points α and β, $\mathbb{B} = \{0, 1\}$ and $\mathbb{B}(x_1, x_2, ..., x_n)$ is the Boolean algebra of all Boolean functions depending of variables $x_1, x_2, ..., x_n$.

Definition 2 [authors (2019)]

Let's call *degree of joint action of the factors* $x_1, x_2, \ldots, x_n$ in a response $f \in \mathbb{B}(x_1, x_2, ..., x_n)$ a number $\mu_f = \max\{\mu_f(\alpha) \mid \alpha \in C_f\}$. For a zero response, $f = 0$, it is convenient to assume that $\mu_f = 0$.

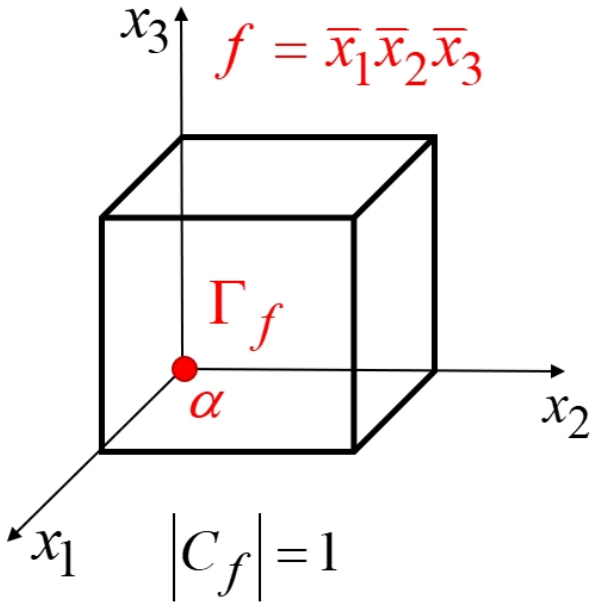
Definition 3 [USEBERIT, 2022]

We call *degree* $\mu_{f,k}$ *of joint action of* k *factors* in a response f the maximal degree of joint action of k factors when arbitrarily fixing the remaining $n - k$ factors and giving them arbitrary values in the response f.

To assess the strength of joint action of the factors in a given response as a whole, let us introduce a set $M_f = (\mu_{f,1}, \mu_{f,2}, \ldots, \mu_{f,n})$.

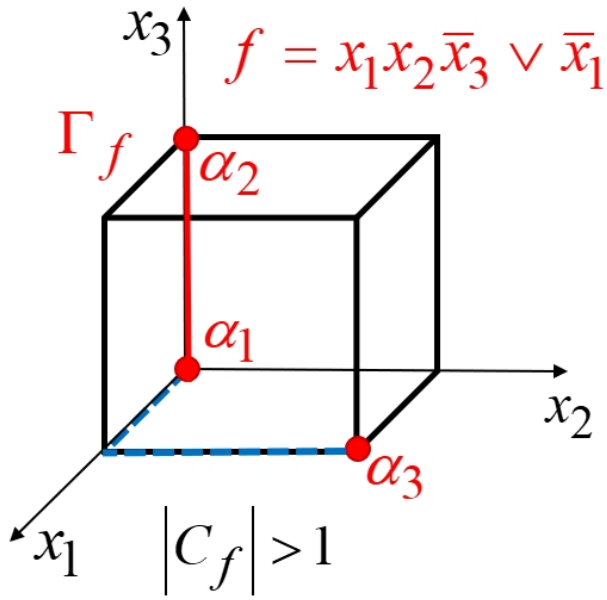
Definition 4

We call M_f the *spectrum* of joint action of the factors in a given response.



$$|C_f| = 1$$

$$\mu_f = \mu_f(\alpha) = n = 3$$



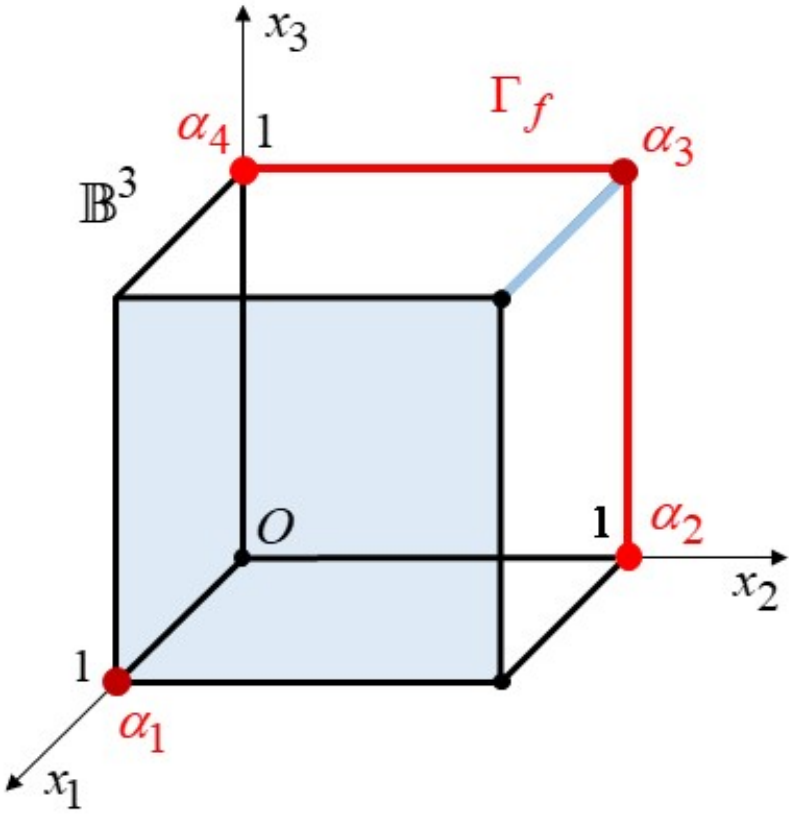
$$|C_f| > 1$$

$$d\left(\alpha_1, C_f \setminus \{\alpha_1\}\right) = 1$$

$$d\left(\alpha_2, C_f \setminus \{\alpha_2\}\right) = 1$$

$$d\left(\alpha_3, C_f \setminus \{\alpha_3\}\right) = 2$$

$$\mu_f = \mu_f(\alpha_3) = 1$$



$$f = x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2 \vee \bar{x}_1x_3$$

$$\mu_{f,3} = \mu_f = \mu_f(\alpha_1) = 1$$

$$\mu_{f,2} = 2 \ \ (x_1 = 1)$$

$$\mu_{f,1} = 1 \ \ (x_2 = 1, x_3 = 1)$$

$$M_f = (1, 2, 1)$$

Let us encode the faces of the Boolean cube $\mathbb{B}^n$ with words w of length n over the alphabet $A = \{*, 0, 1\}$ as it is generally accepted in the theory of Boolean functions.

One can consider the word w as a ternary notation ω of a decimal integer from 0 up to $3^n - 1$, where we set up the following matching of the characters from the alphabet A to symbols $\{0, 1, 2\}$: $* \leftrightarrow 0, 0 \leftrightarrow 1,\ 1 \leftrightarrow 2$.

Example 1. For the record $\omega = 002$ a related word is $w = **1$ and corresponding 2-face is specified by the condition $x_3 = 1$.
For the record $\omega = 201$ a related word is $w = 1*0$ and corresponding 1-face is specified by the conditions $x_1 = 1$, $x_3 = 0$.

Let Ω denote an ordered set of all the ternary notations ω.

The set Ω is linearly ordered with respect to the natural lexicographic order $\preccurlyeq$.

Denote by 0_n the zero tuple of length n and $\Omega_0 = \Omega \setminus \{0_n\}$.

For any word $\omega = \omega_1\omega_2 \dots \omega_n$ from Ω, denote $t(\omega)$ the index of the last nonzero character in the word ω if $\omega \neq 0_n$, and $t(\omega) = 0$ if $\omega = 0_n$.

Example 2. Let $n = 3$. For a word $\omega = 120$ we have $t(\omega) = 2$.
For a word $\omega = 000$ we have $t(\omega) = 0$.

For each word $\omega \in \Omega_0$ we denote by $\varphi(\omega)$ the word obtained from the word ω by replacing the character with the index $t(\omega)$ with a zero character.

Example 3. Let $n = 4$. For a word $\omega = 0210$ we have $\varphi(\omega) = 0200$; and for a word $\omega = 1102$ we have $\varphi(\omega) = 1100$.

Consider a directed graph $T = (\Omega, \rho)$, where $\rho = \varphi^{-1}$ is a converse relation of the binary relation φ. The next statements hold.

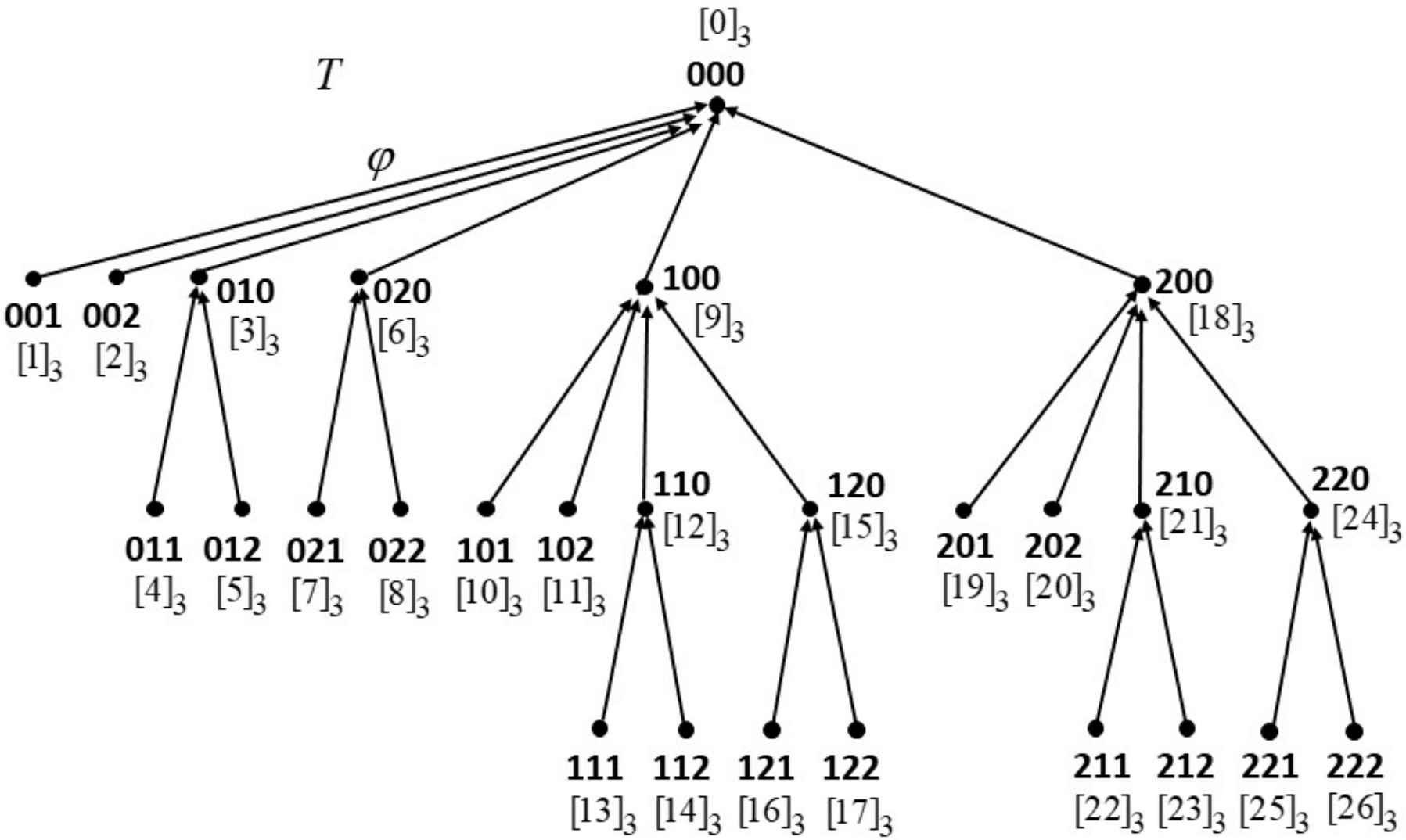
Theorem 1

The graph T is a directed tree with the root 0_n, height n, and the node $\varphi(\omega)$ is a parent of the node $\omega \in \Omega_0$.

Theorem 2

Preorder on the nodes of the tree T coincides with the order $\preccurlyeq$ on the set Ω.

Slide 8. **Geometrical illustration of the tree T**



The Algorithm 3 is based on a nonrecursive preorder traversal of the tree T nodes. It is a modification of the Algorithm 1 created in view of the Theorem 2.

To speed up the program, the vertices of the Boolean cube $\mathbb{B}^n$ are encoded by their decimal representations, and bitwise XOR is used to calculate the Hamming distance between the vertices of the Boolean cube.

A geometrical illustration of Algorith 3 execution and its pseudocode are given below.

Lemma 1. The time complexity of the Algorithm 3 is no more than $O(6^n)$.

Lemma 2. The time complexity of any algorithm which calculates the interaction spectrum by definition is no less than $O(6^n)$.

From Lemmas 1 and 2 follows the

Theorem 3

The time complexity of the Algorithm 3 is equal to $O(6^n)$ and it is optimal among all the algorithms that calculate the spectrum of the joint action by definition.

n	Execution time: min : sec : ms		
	Algorithm 1	Algorithm 2	Algorithm 3
2	00:00:000	00:00:030	00:00:003
3	00:00:005	00:00:030	00:00:003
4	00:00:005	00:00:033	00:00:003
5	00:00:007	00:00:033	00:00:003
6	00:00:012	00:00:036	00:00:011
7	00:00:023	00:00:051	00:00:016
8	00:00:056	00:00:105	00:00:024
9	00:00:160	00:00:395	00:00:059
10	00:00:661	00:01:519	00:00:167
11	00:03:302	00:06:933	00:00:718
12	00:18:032	00:28:766	00:04:305
13	02:19:797	02:15:426	00:28:481
14	09:40:696	—	02:41:695
15	58:21:365	—	16:55:709

The Algorithm 2, whose upper estimate of time complexity is less than that of the Algorithm 1, is supposed to outpace the Algorithm 1 for large n.

From this table we see that the Algorithm 2 outpaces the Algorithm 1 at $n = 13$.

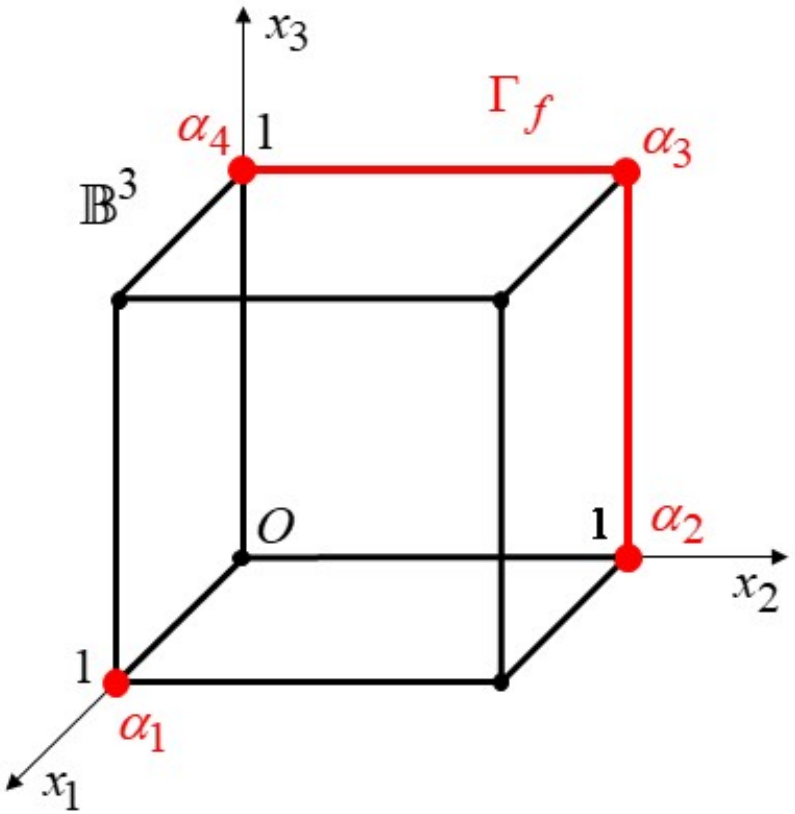
But already at $n = 14$, $n = 15$ the execution time of Algorithm 2 becomes unacceptably long on a personal computer with a usual hardware due to the lack of RAM and persistent access to the swap file.

Meanwhile, Algorithm 3 works fastest at all the values n under consideration.

Data: Boolean function f
Result: Mf — spectrum of the Boolean function f
Program Algorithm 3
const n: *integer;*
var
S : *stack of* Ω; //S is the stack holding the path from the root to the parent $top(S)$ of the current node ω;
C : *stack of Set*; //C is stack of sets $C(\omega)$, $\omega \in S$;
m : *array* $[1..n]$ *of integer*; //m is the dynamically generated spectrum M_f;
k : *integer*; $\omega : \bar{\Omega}$; D : *Set*; //k, ω, D are temporary variables;
function $Mu(D : Set)$: *integer*; //Calculate the degree of the joint action over the set D
function $CreateSet(\omega : \Omega)$: *Set*; //Create the set $C(\omega)$ for an $\omega \in \Omega_0$. by the taking into account that $C(\varphi(\omega)) = top(C)$
function $MinChild(\omega : \Omega) : \bar{\Omega}$. //If ω is not a leaf, then find the minimal child of ω and return λ if ω is a leaf.
function $MinRightSib(\omega : \Omega) : \bar{\Omega}$; //Find the minimal right sibling of ω, if it has a right siblings, and return λ otherwise.

begin
  for $k := 1$ **to** n **do** $m_k := 0$;
  $S := \varnothing$; $C := \varnothing$;
  $k := n$;
  $m[n] := Mu(C_f)$; //Calculation of the degree $\mu_{f,n} = \mu_f$ by the Definit. 2,3
  $S \leftarrow 0_n$; $C \leftarrow C_f$;
  $\omega := MinChild(0_n)$; //Explore minimal child of the node 0_n.
  $k := n - 1$;
  while $S \neq \varnothing$ **do** //While not all the nodes are traversed.
    begin
      if $\omega \neq \lambda$ **then**
        begin
          $D := CreateSet(\omega)$; //Create the set $C(\omega)$.
          if $k > 0$ **then** $m[k] := \max\{m[k], Mu(D)\}$;
          //Dynamical calculation of the degree $\mu_{f,k}$ by the Definition 5.
          $S \leftarrow \omega$; $C \leftarrow D$;
          $\omega := MinChild(\omega)$; //Explore the minimal child of the node ω
          $k := k - 1$
        end

while $S \neq \varnothing$ **do** //While not all the nodes are traversed.
begin
if $\omega \neq \lambda$ **then**
begin
$D := CreateSet(\omega)$; //Create the set $C(\omega)$.
if $k > 0$ **then** $m[k] := \max\{m[k], Mu(D)\}$;
//Dynamical calculation of the degree $\mu_{f,k}$ by the Definition 5.
$S \leftarrow \omega$; $C \leftarrow D$;
$\omega := MinChild(\omega)$; //Explore the minimal child of the node ω
$k := k - 1$
end
else
begin
$\omega := MinRightSib\,(top(S))$;
$k := k + 1$;
$POP(S)$; $POP(C)$//Pop the node $top(S)$ out of the
stack S and the set $top(C)$ out of the stack C.
end
end
end

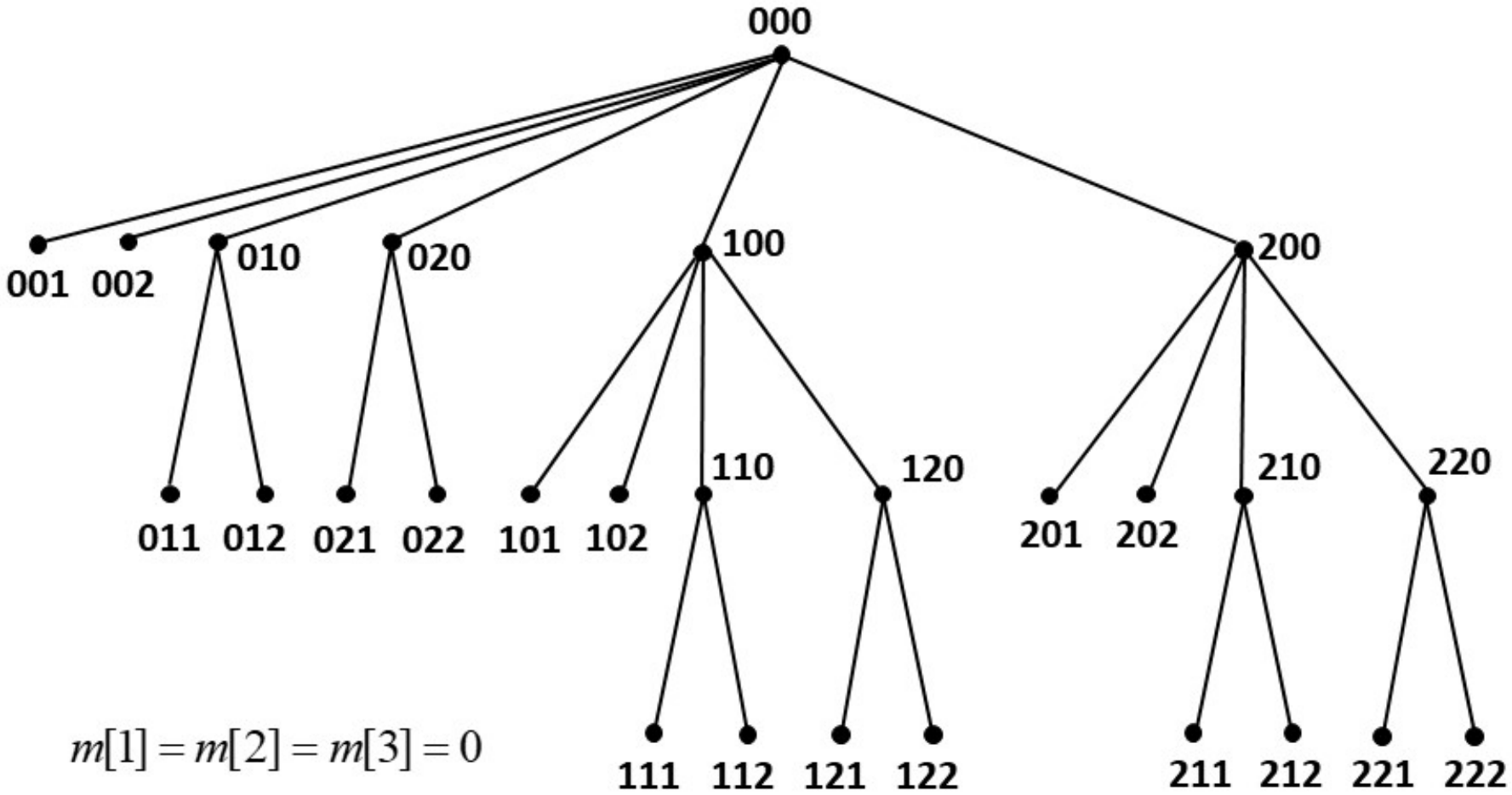


$$n = 3$$

$$f = x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2 \vee \bar{x}_1x_3$$

$$C_f = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$$

Slide 16. Geometrical illustration of the program execution

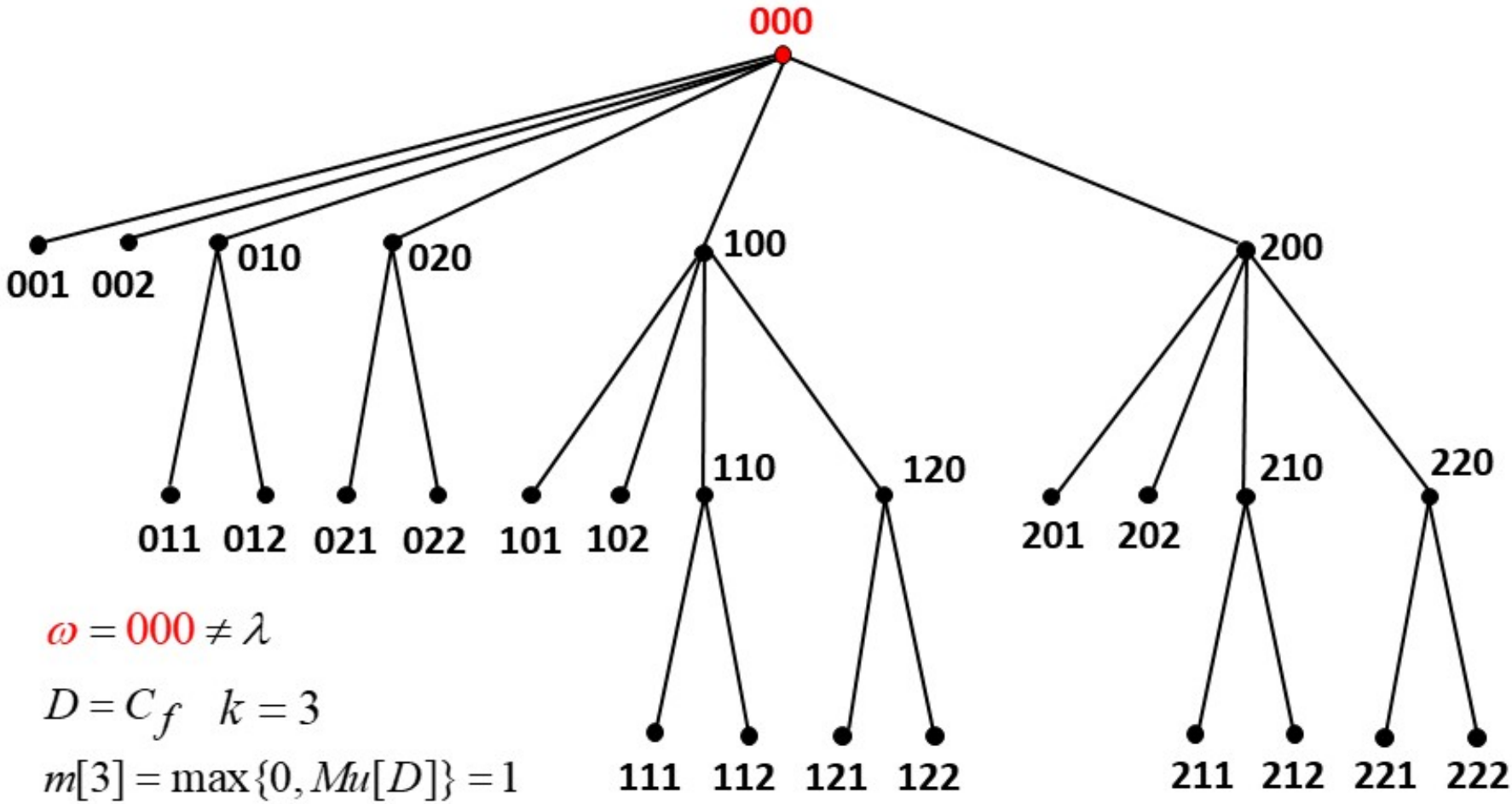


$m[1] = m[2] = m[3] = 0$

$S = \varnothing \quad C = \varnothing$

$\boldsymbol{m = (0, 0, 0)}$

Slide 16. Geometrical illustration of the program execution



$\omega = 000 \neq \lambda$

$D = C_f \quad k = 3$

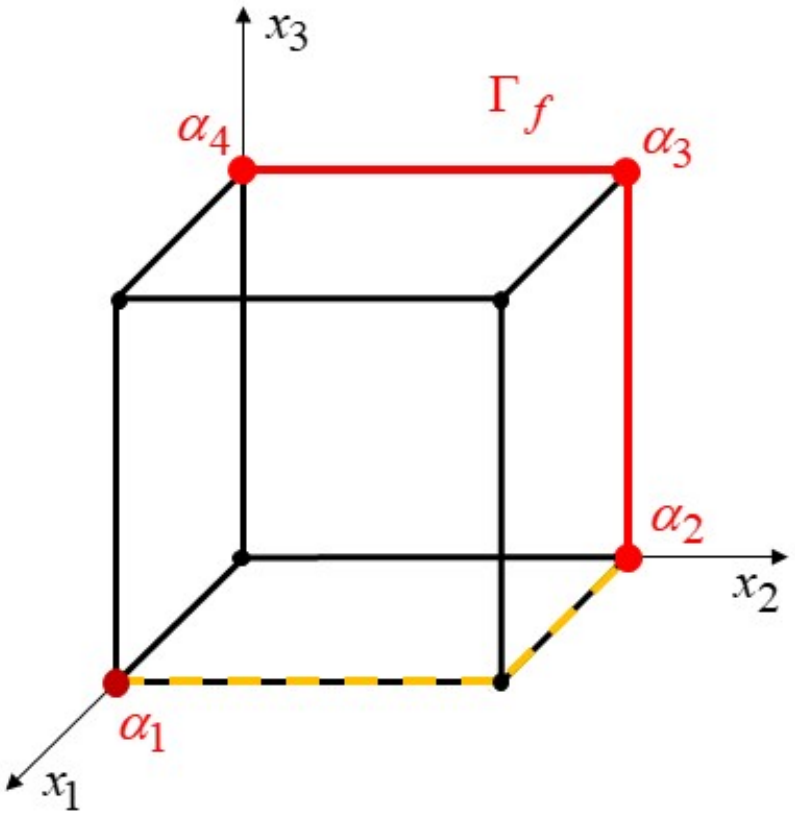
$m[3] = \max\{0, Mu[D]\} = 1$

$S \leftarrow \omega \quad C \leftarrow D$

$S = \{000\} \quad C = \{C_f\}$

$\omega = MinChild(\omega) = 001$

$\boldsymbol{m = (0, 0, 1)}$



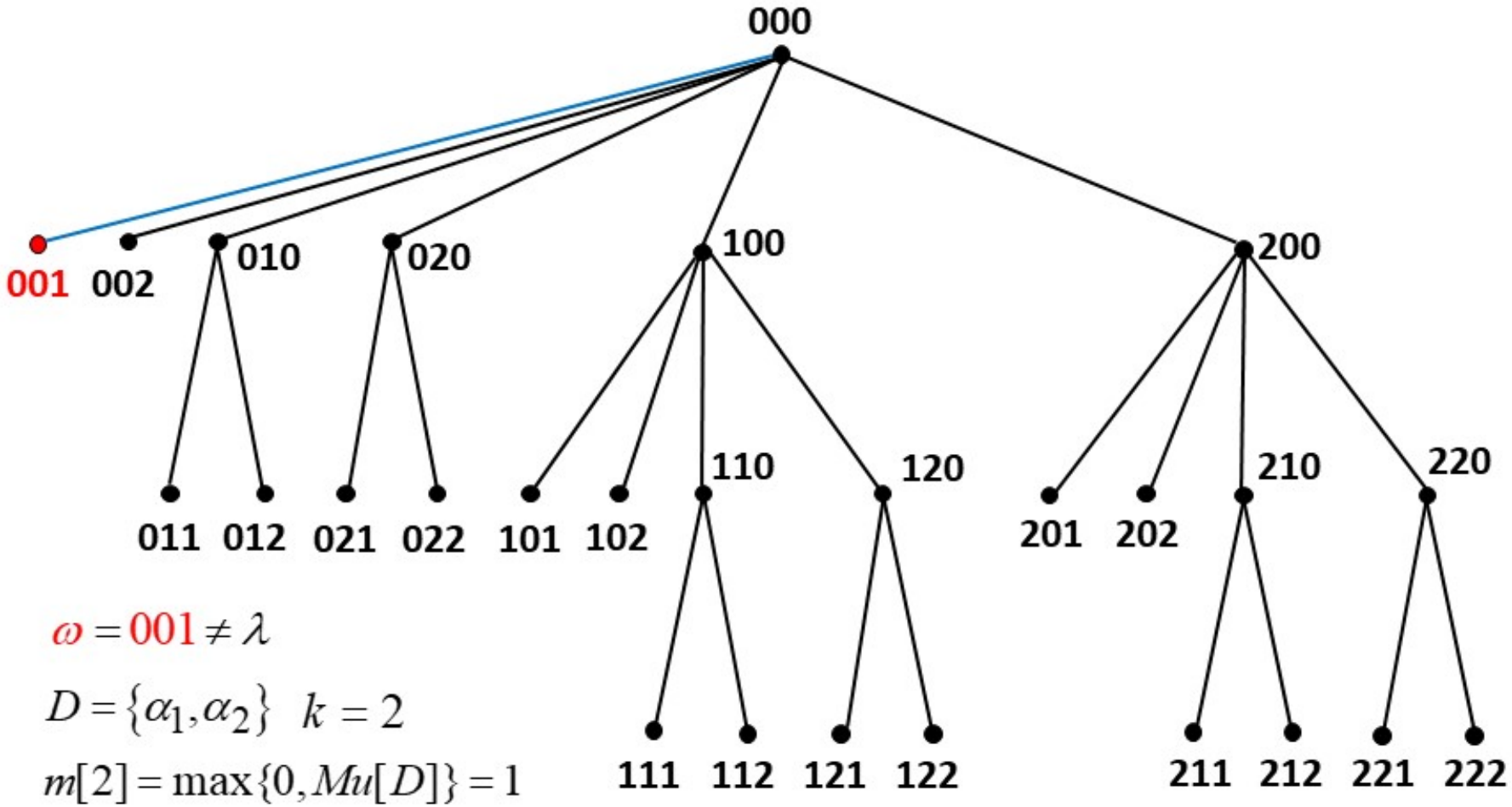
$$\omega = 000$$

$$w = ***$$

$$D = C_f$$

$$Mu(D) = d(\alpha_1, D \setminus \{\alpha_1\}) - 1 = 1$$

Slide 17. Geometrical illustration of the program execution



$\omega = 001 \neq \lambda$

$D = \{\alpha_1, \alpha_2\} \quad k = 2$

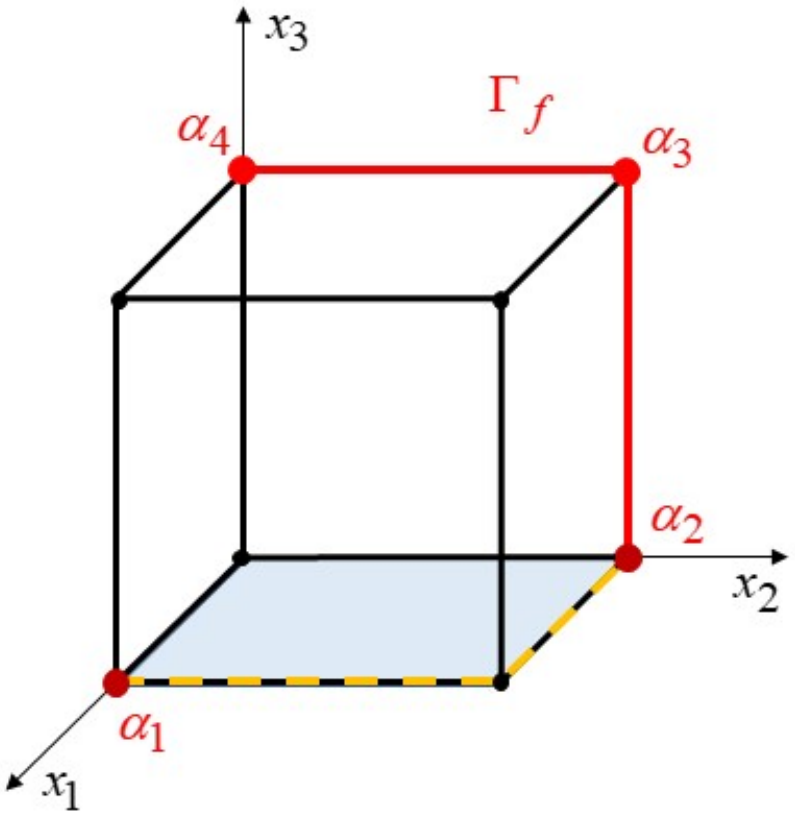
$m[2] = \max\{0, Mu[D]\} = 1$

$S \leftarrow \omega \quad C \leftarrow D$

$S = \{000, 001\} \quad C = \{C_f, \{\alpha_1, \alpha_2\}\}$

$\omega = MinChild(\omega) = \lambda$

$\boldsymbol{m = (0, 1, 1)}$



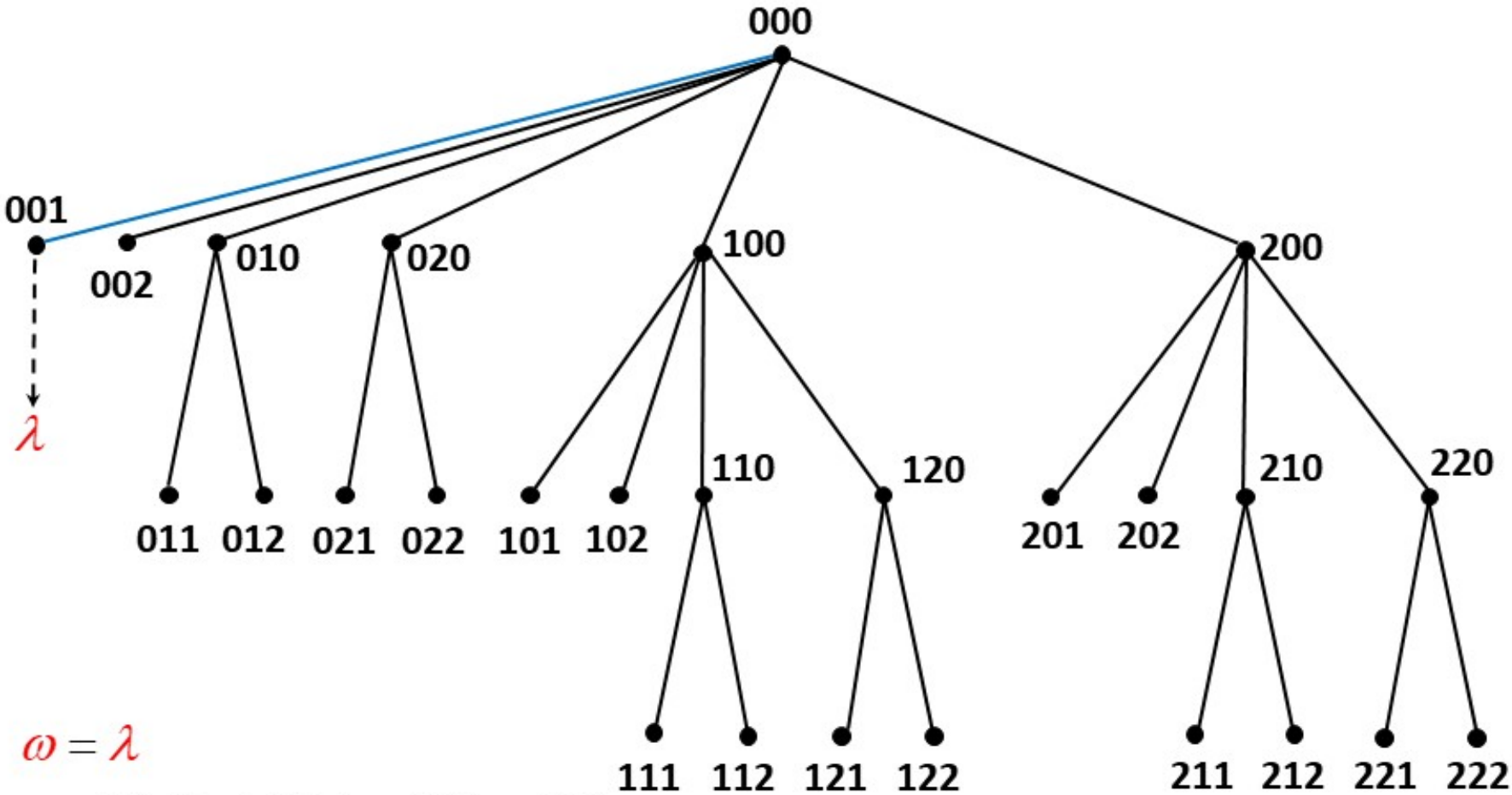
$$\omega = 001$$

$$w = **0 \quad x_3 = 0$$

$$D = \{\alpha_1, \alpha_2\}$$

$$Mu(D) = d(\alpha_1, D \setminus \{\alpha_1\}) - 1 = 1$$

Slide 17. Geometrical illustration of the program execution



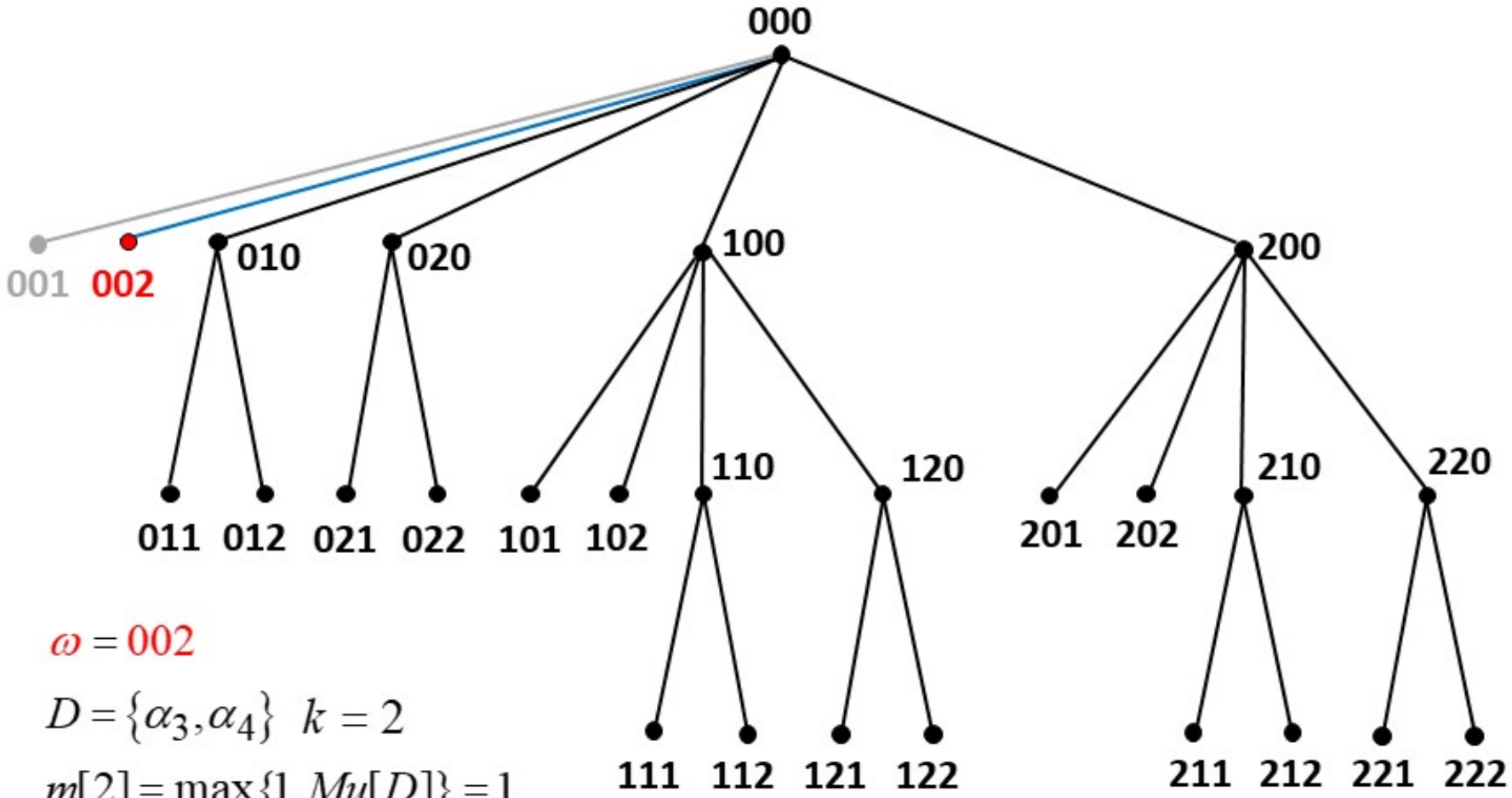
$\omega = \lambda$

$\omega = MinRightSib(top(S)) = 002$

$S \to top(S) \quad C \to top(C)$

$S = \{000\} \quad C = \{C_f\}$

Slide 17. Geometrical illustration of the program execution



$\omega = 002$

$D = \{\alpha_3, \alpha_4\} \quad k = 2$

$m[2] = \max\{1, Mu[D]\} = 1$

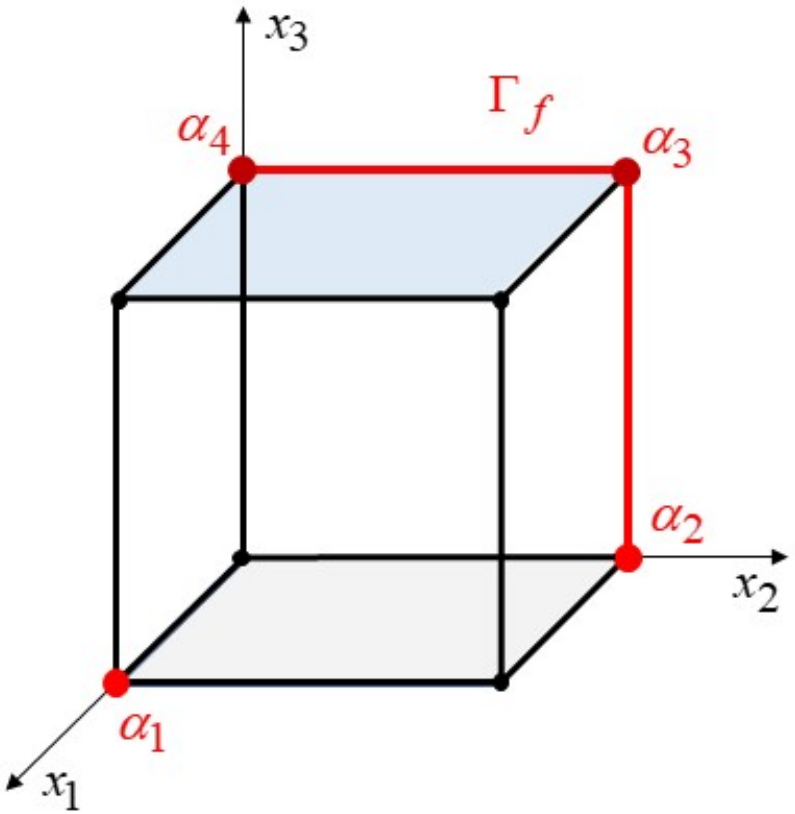
$S \leftarrow \omega \quad C \leftarrow D$

$S = \{000, 002\} \quad C = \{C_f, \{\alpha_3, \alpha_4\}\}$

$\omega = MinChild(\omega) = \lambda$

$\boldsymbol{m = (0, 1, 1)}$

Slide 18. Geometrical illustration of the program execution



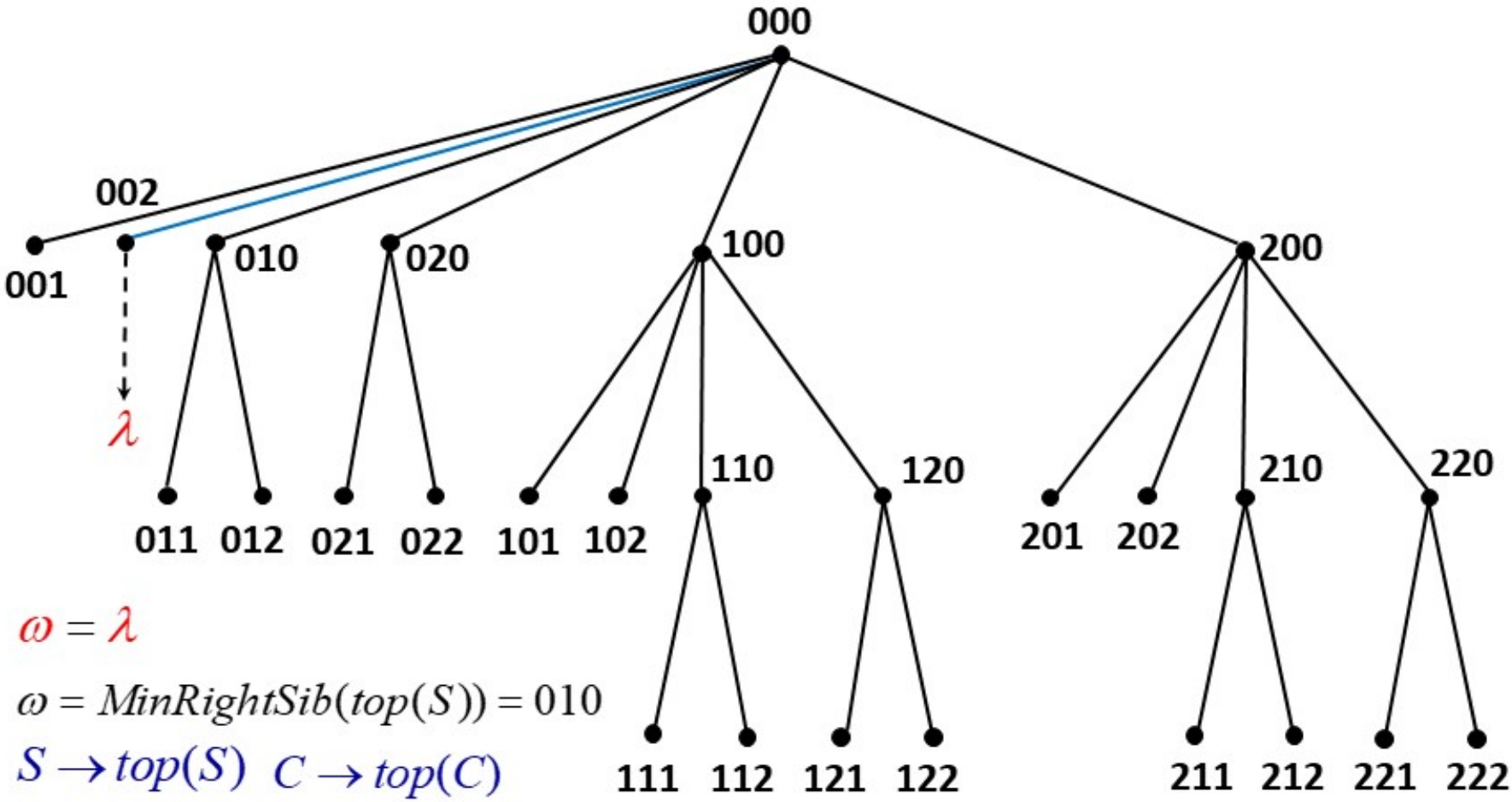
$$\omega = 002$$

$$w = **1 \quad x_3 = 1$$

$$D = \{\alpha_3, \alpha_4\}$$

$$Mu(D) = d(\alpha_3, D \setminus \{\alpha_3\}) - 1 = 0$$

Slide 18. Geometrical illustration of the program execution



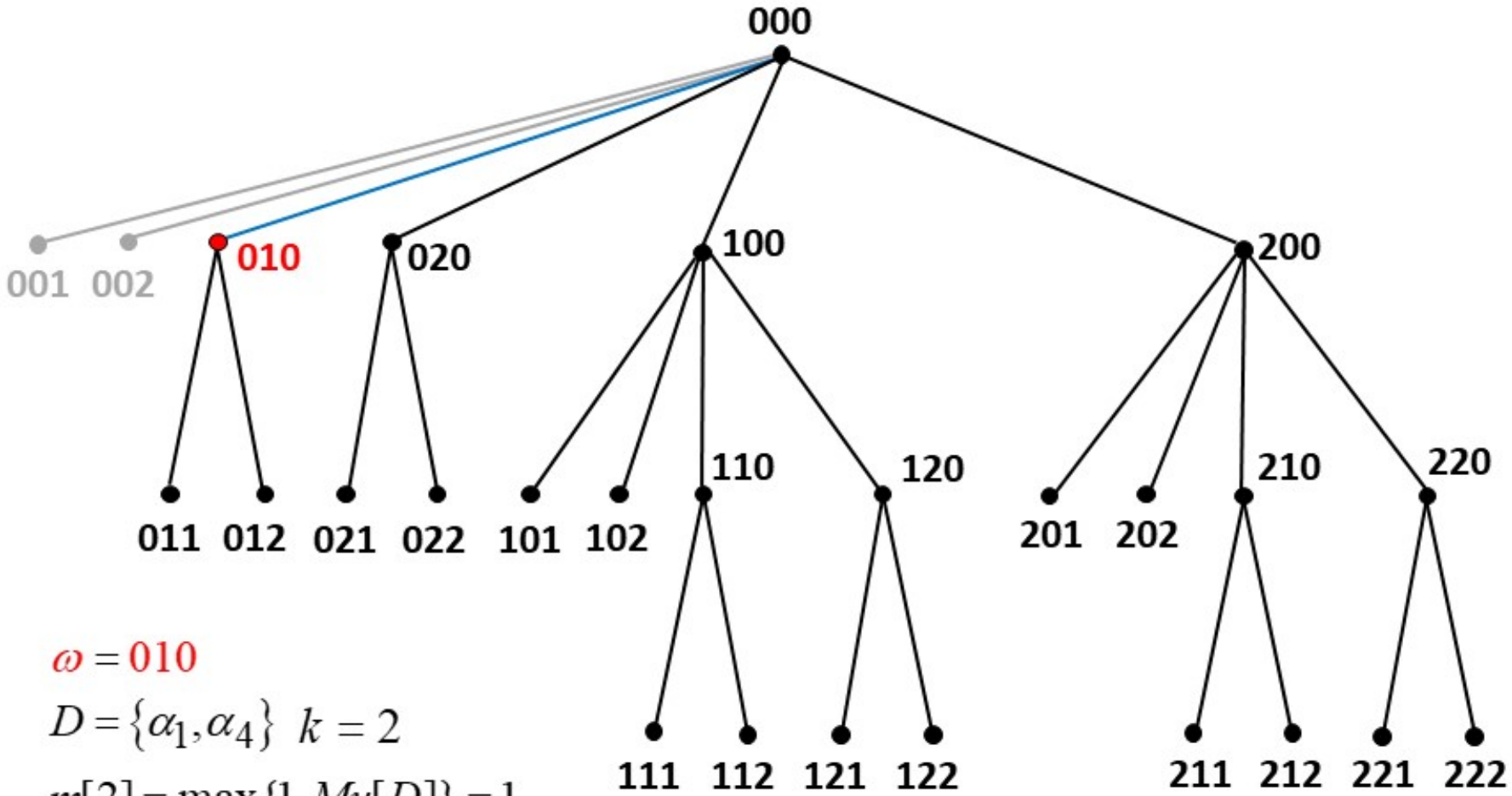
$\omega = \lambda$

$\omega = MinRightSib(top(S)) = 010$

$S \rightarrow top(S) \quad C \rightarrow top(C)$

$S = \{000\} \quad C = \{C_f\}$

Slide 18. Geometrical illustration of the program execution



$\omega = 010$

$D = \{\alpha_1, \alpha_4\} \quad k = 2$

$m[2] = \max\{1, Mu[D]\} = 1$

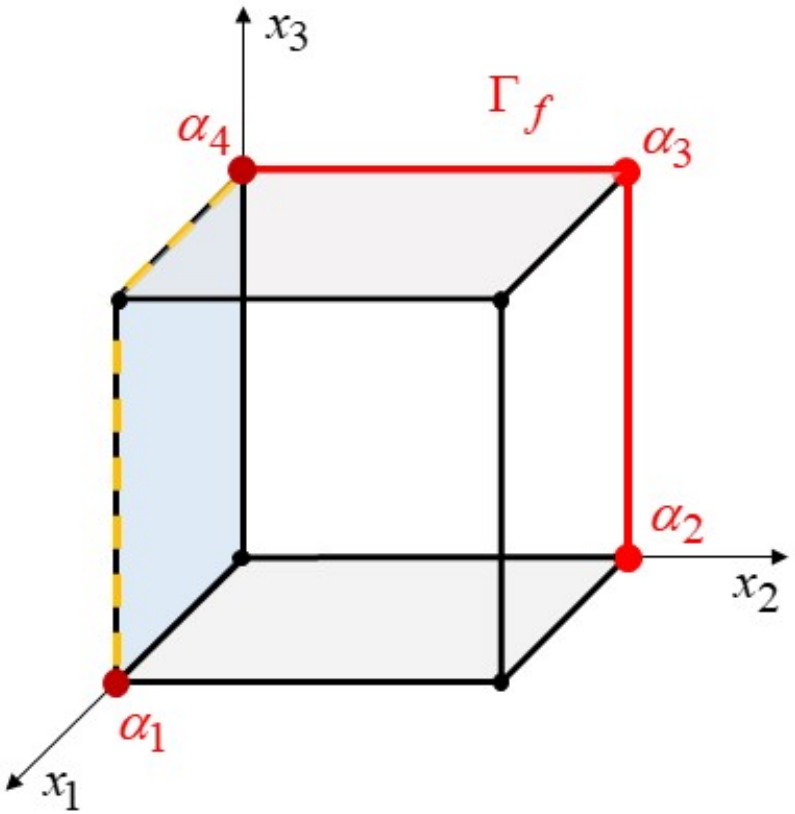
$S \leftarrow \omega \quad C \leftarrow D$

$S = \{000, 010\} \quad C = \{C_f, \{\alpha_1, \alpha_4\}\}$

$\omega = MinChild(\omega) = 011$

$\boldsymbol{m = (0, 1, 1)}$

Slide 18. Geometrical illustration of the program execution



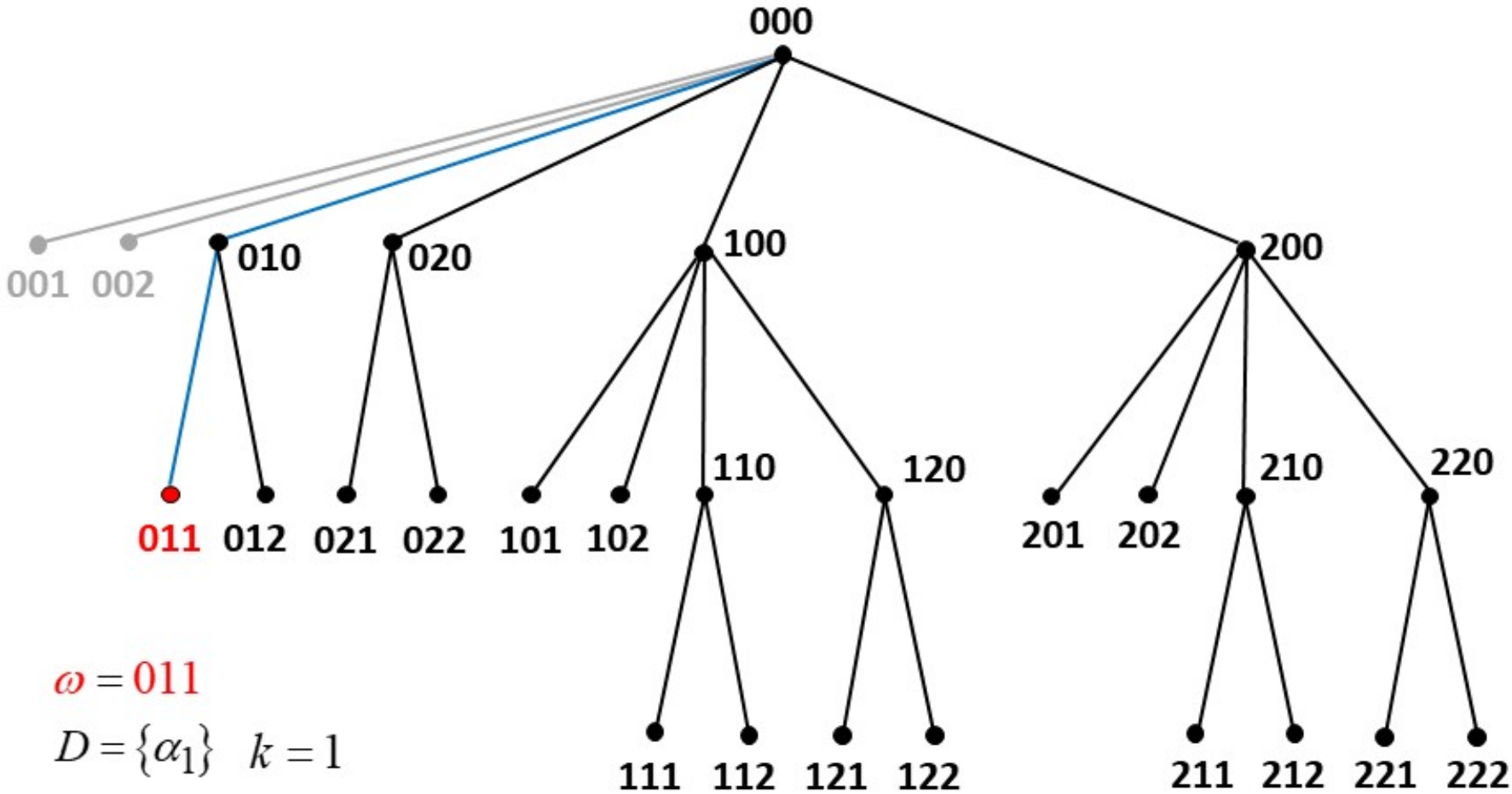
$\omega = 010$

$w = *0*$ $x_2 = 0$

$D = \{\alpha_1, \alpha_4\}$

$Mu(D) = d(\alpha_1, D \setminus \{\alpha_1\}) - 1 = 1$

Slide 19. Geometrical illustration of the program execution



$\omega = 011$

$D = \{\alpha_1\} \quad k = 1$

$m[1] = \max\{0, Mu[D]\} = 1$

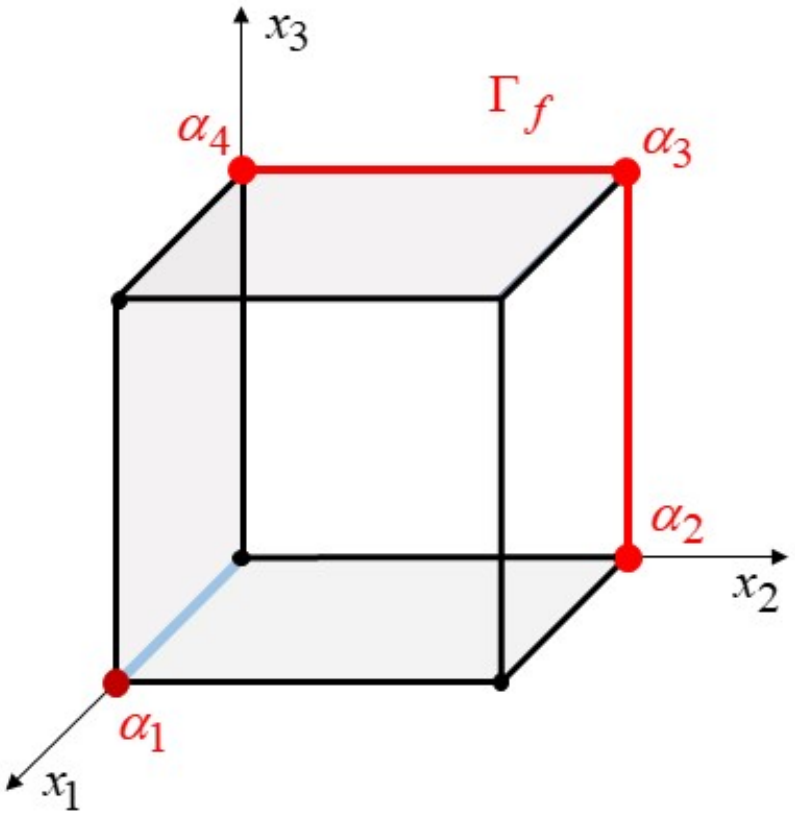
$S \leftarrow \omega \quad C \leftarrow D$

$S = \{000, 010, 011\} \quad C = \{C_f, \{\alpha_1, \alpha_4\}, \{\alpha_1\}\}$

$\boldsymbol{m = (1,1,1)}$

$\boldsymbol{\omega = MinChild(\omega) = \lambda}$

Slide 19. Geometrical illustration of the program execution



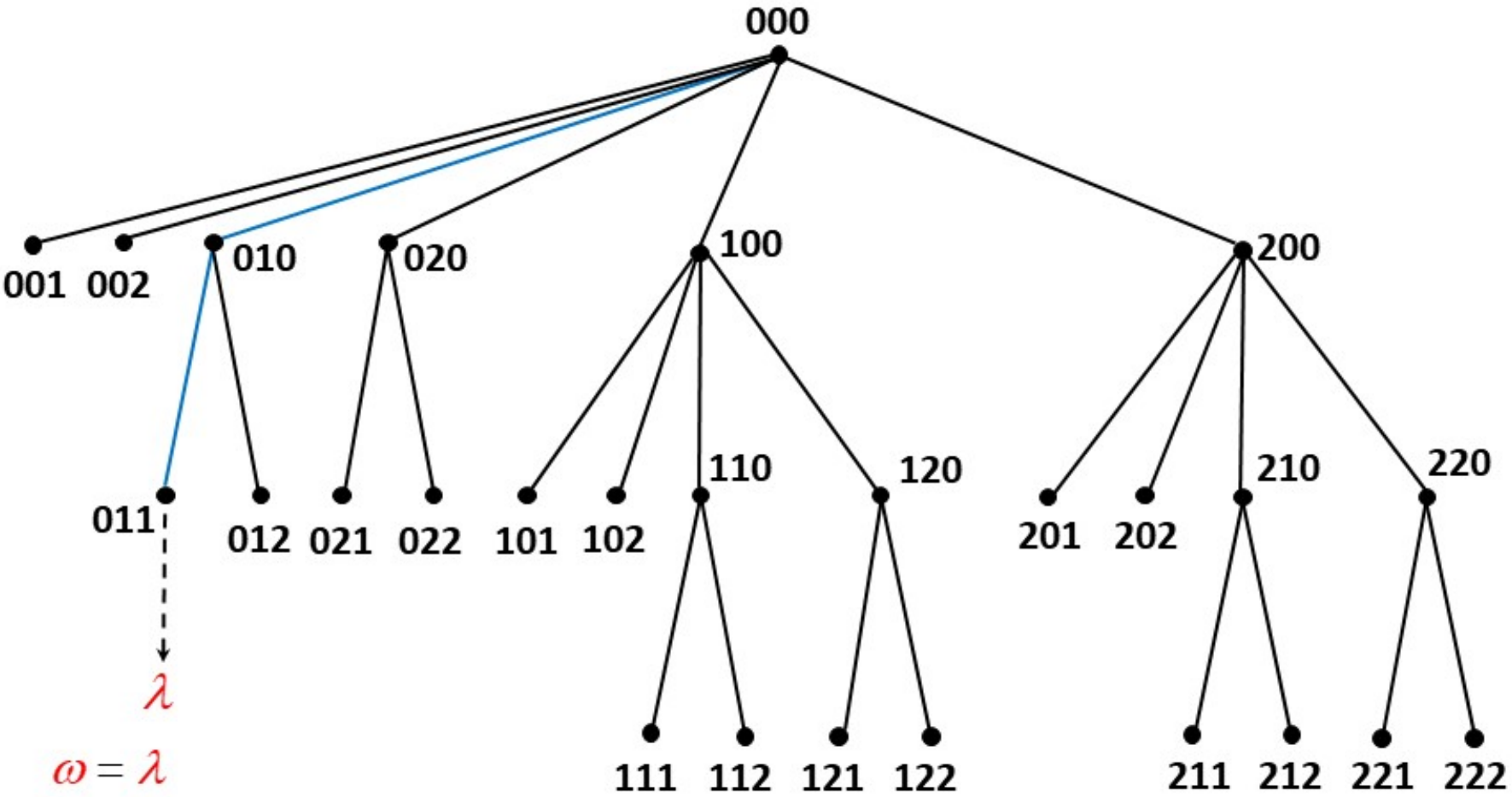
$\omega = 011$

$w = *00 \quad x_2 = 0, x_3 = 0$

$D = \{\alpha_1\}$

$Mu(D) = 1$

Slide 19. Geometrical illustration of the program execution



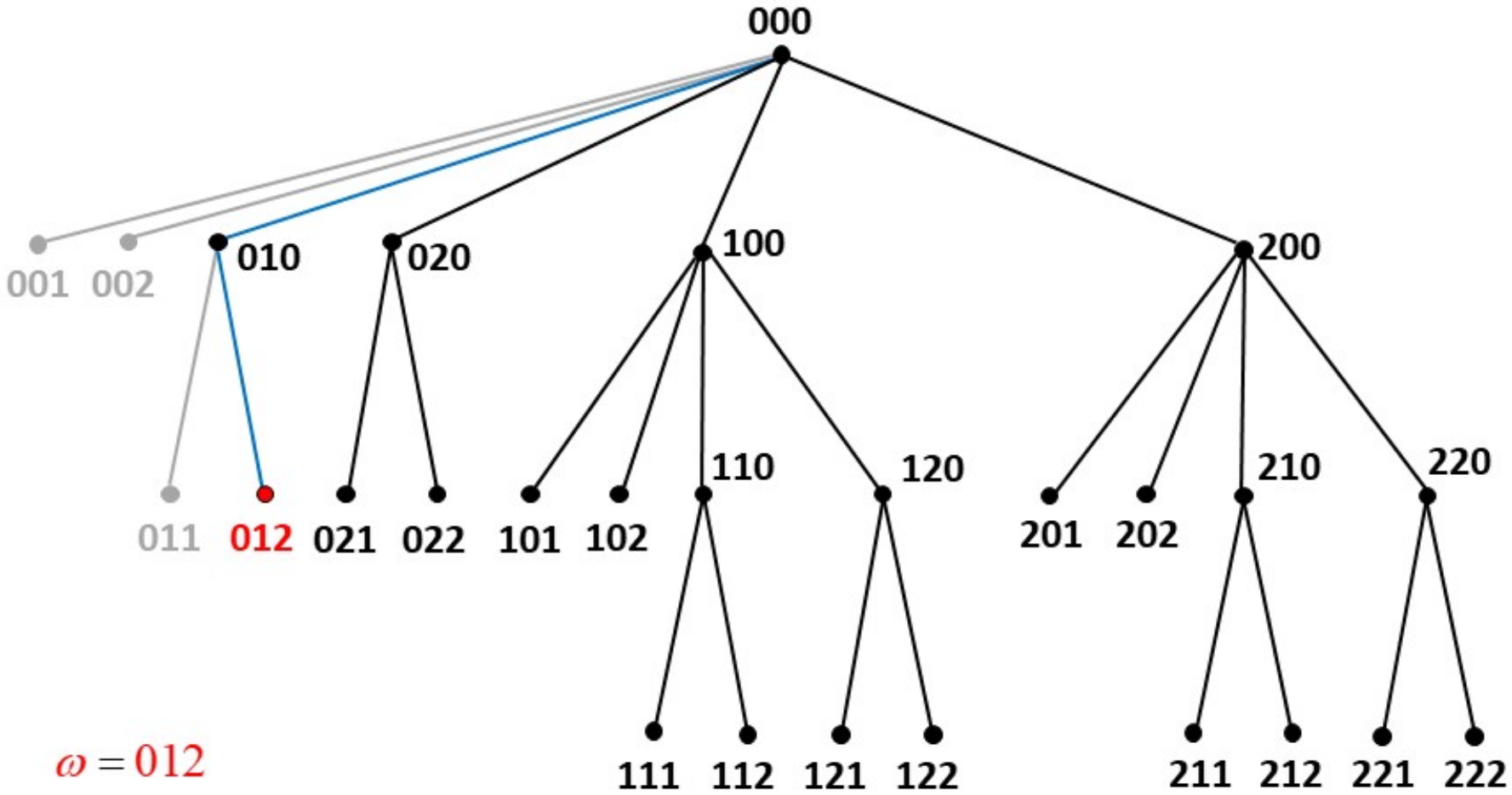
$\omega = \lambda$

$\omega = MinRightSib(top(S)) = 012$

$S \to top(S) \quad C \to top(C)$

$S = \{000, 010\} \quad C = \{C_f, \{\alpha_1, \alpha_4\}\}$

Slide 19. Geometrical illustration of the program execution



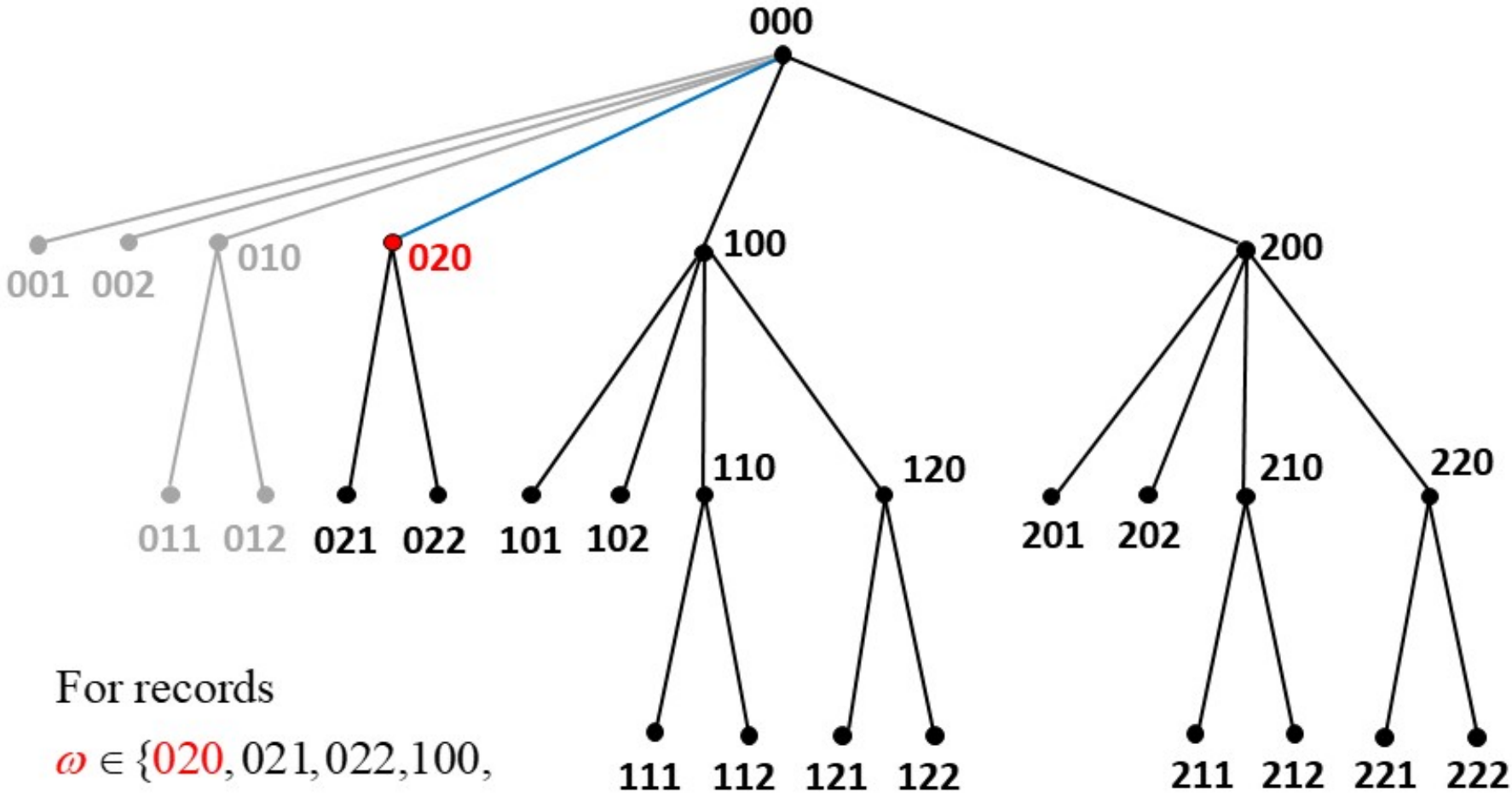
$\omega = 012$

Similarly,

$m[1] = \max\{1, Mu[D]\} = 1$

$\boldsymbol{m = (1, 1, 1)}$

Slide 20. **Geometrical illustration of the program execution**

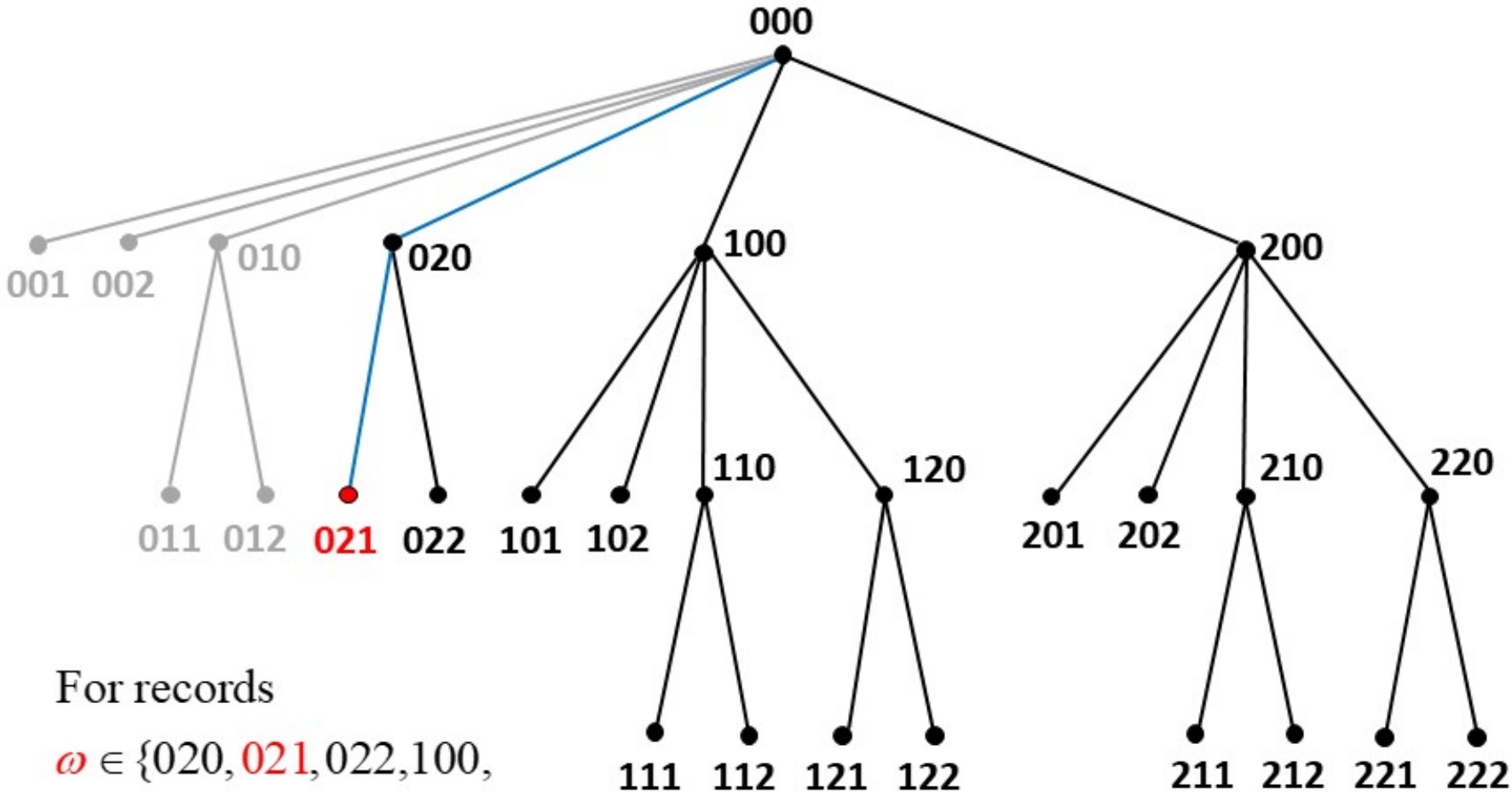


For records

$\omega \in \{020, 021, 022, 100,$
$101, 102, 110, 111,$
$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 20. **Geometrical illustration of the program execution**

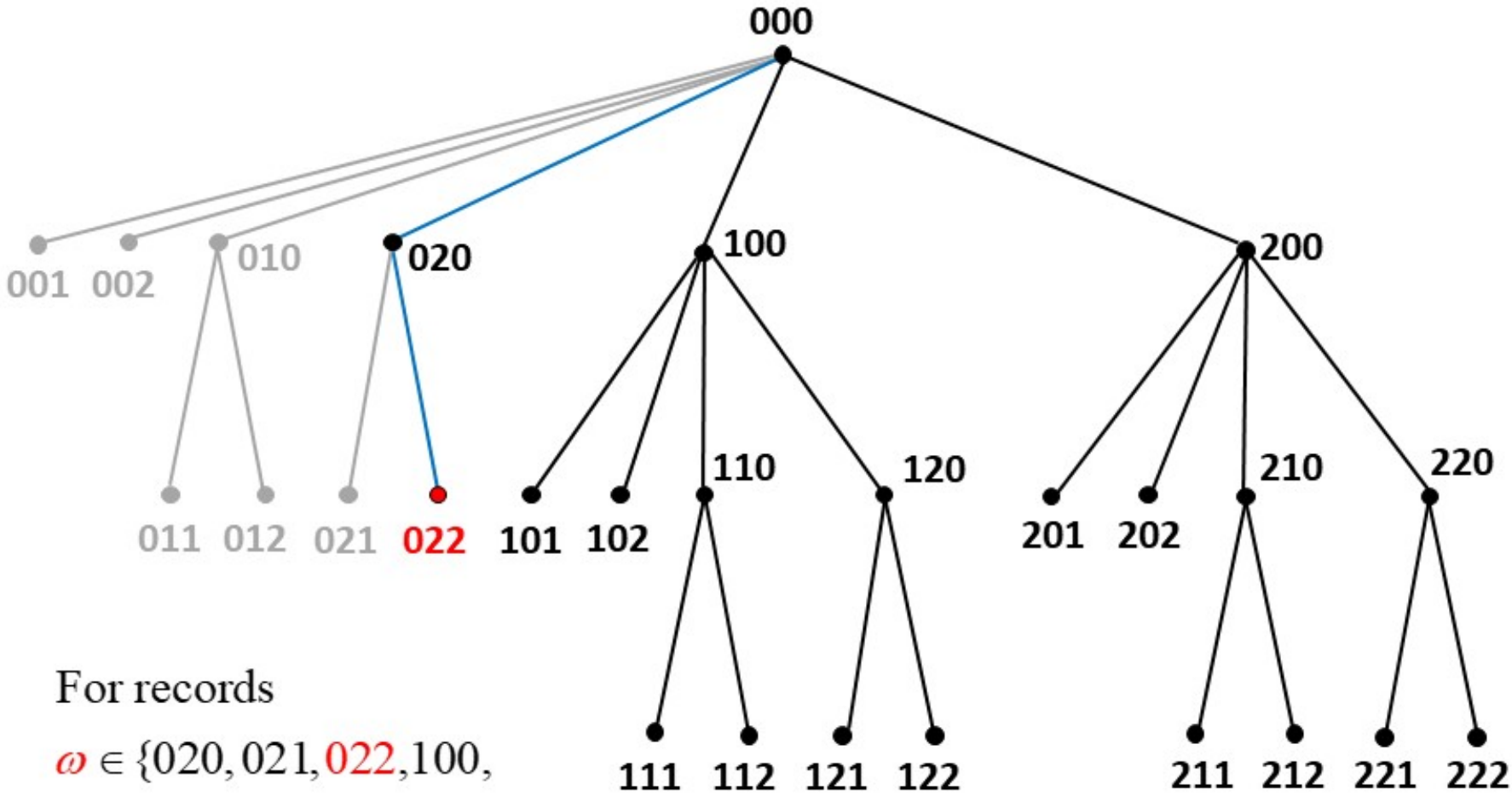


For records

$\omega \in \{020, 021, 022, 100,$
$101, 102, 110, 111,$
$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 20. **Geometrical illustration of the program execution**

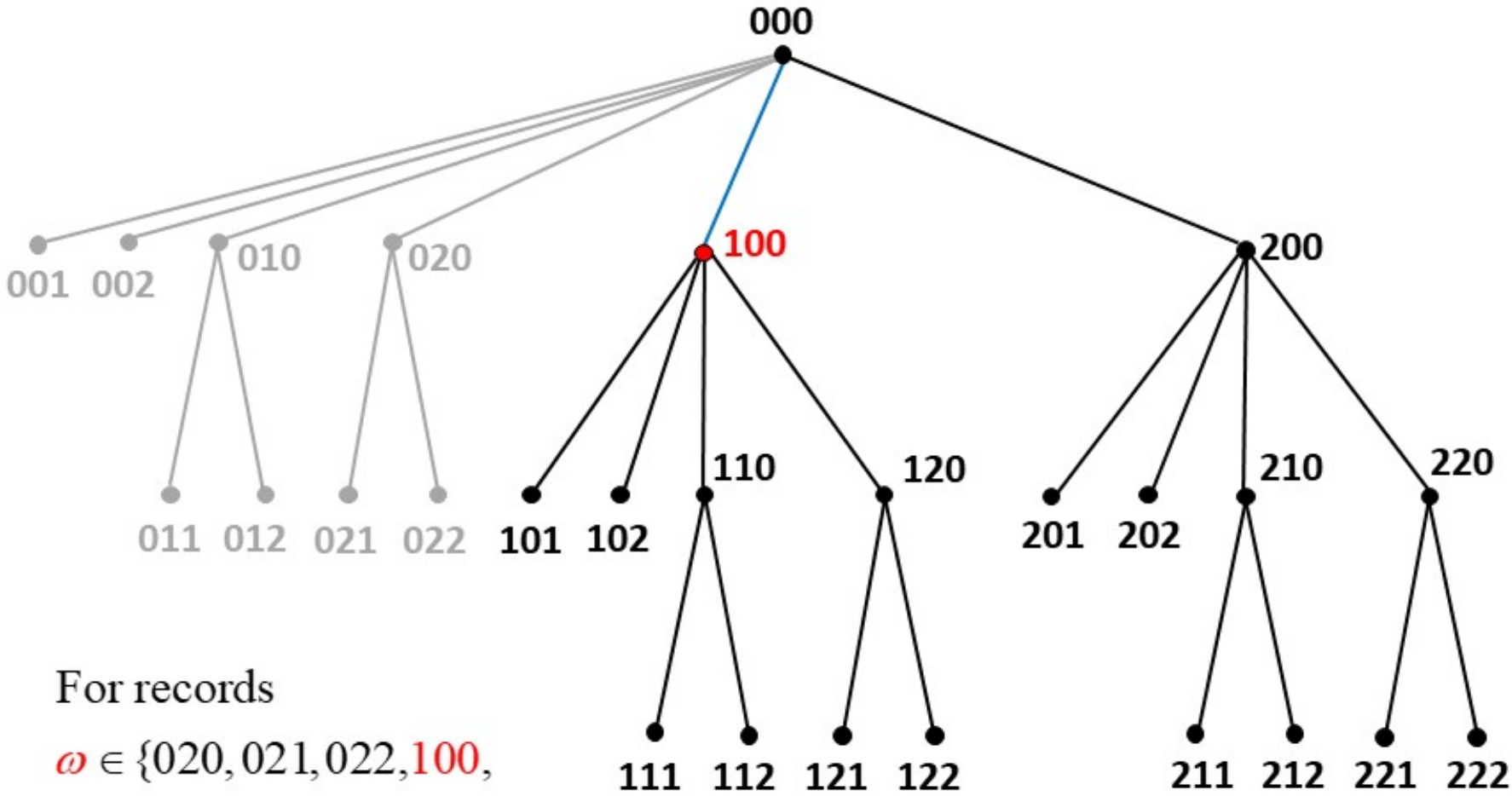


For records

$\omega \in \{020, 021, 022, 100,$
$101, 102, 110, 111,$
$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 20. Geometrical illustration of the program execution



For records

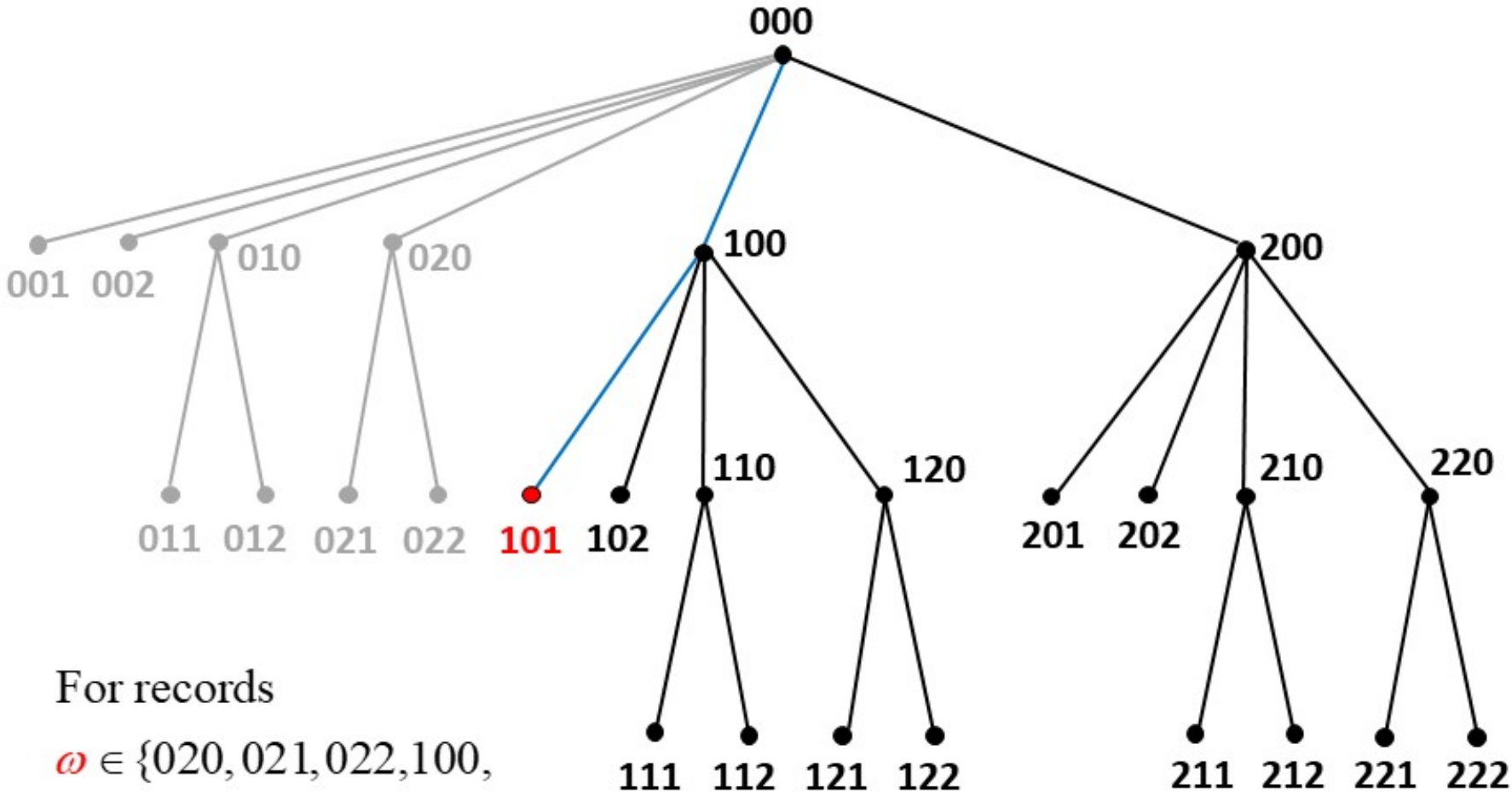
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 21. Geometrical illustration of the program execution



For records

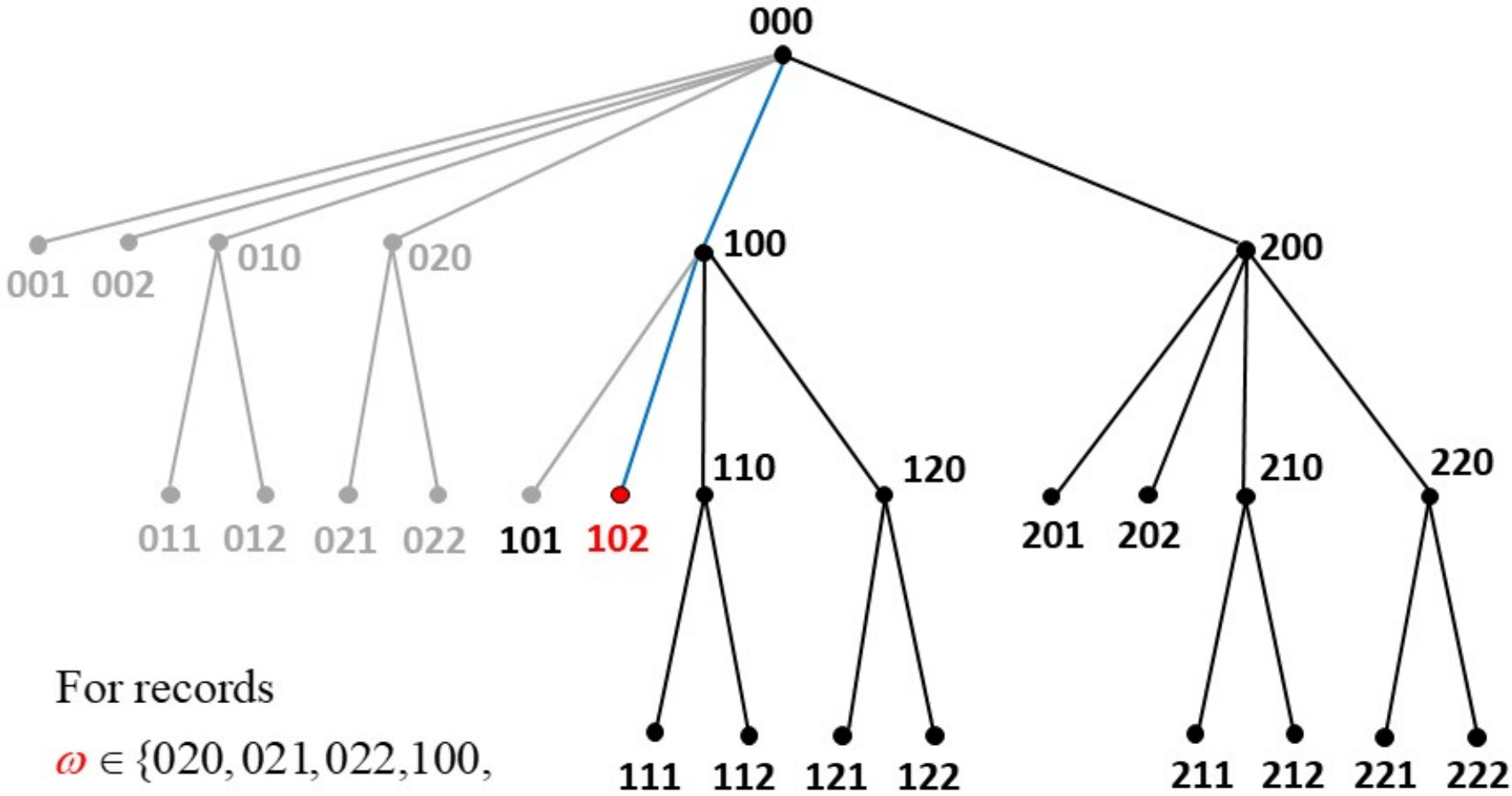
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 21. Geometrical illustration of the program execution



For records

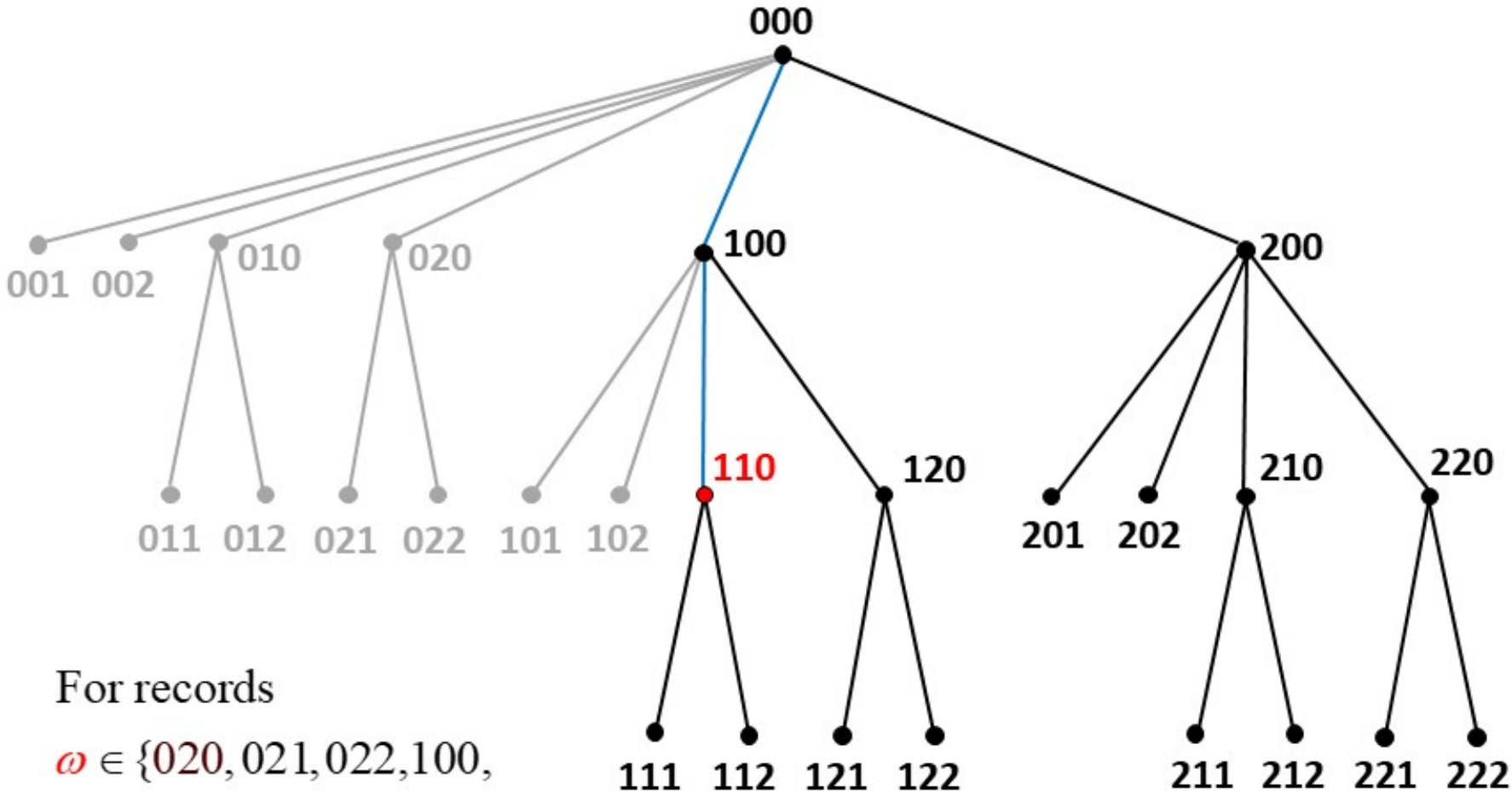
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 21. Geometrical illustration of the program execution



For records

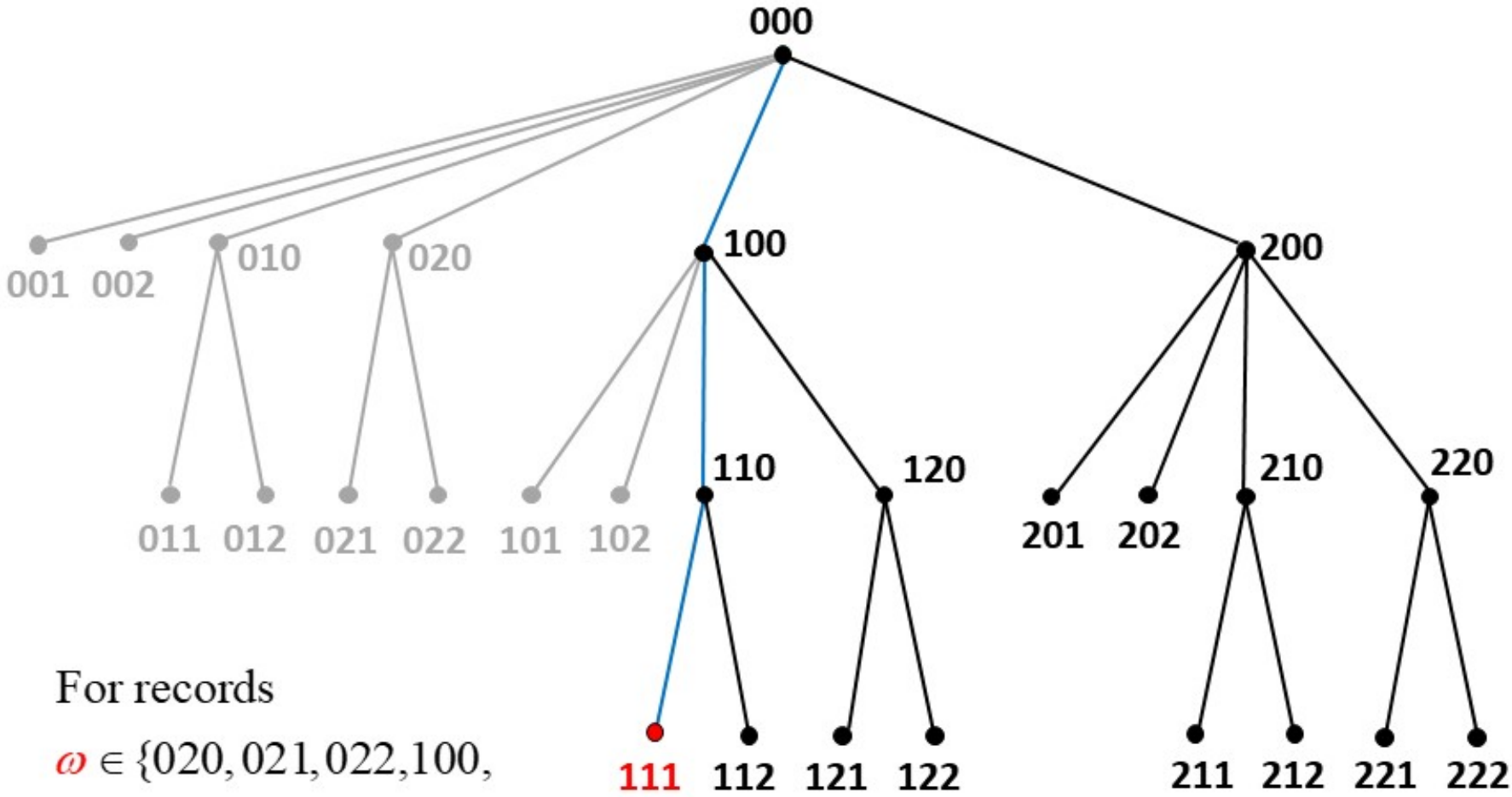
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 22. **Geometrical illustration of the program execution**



For records

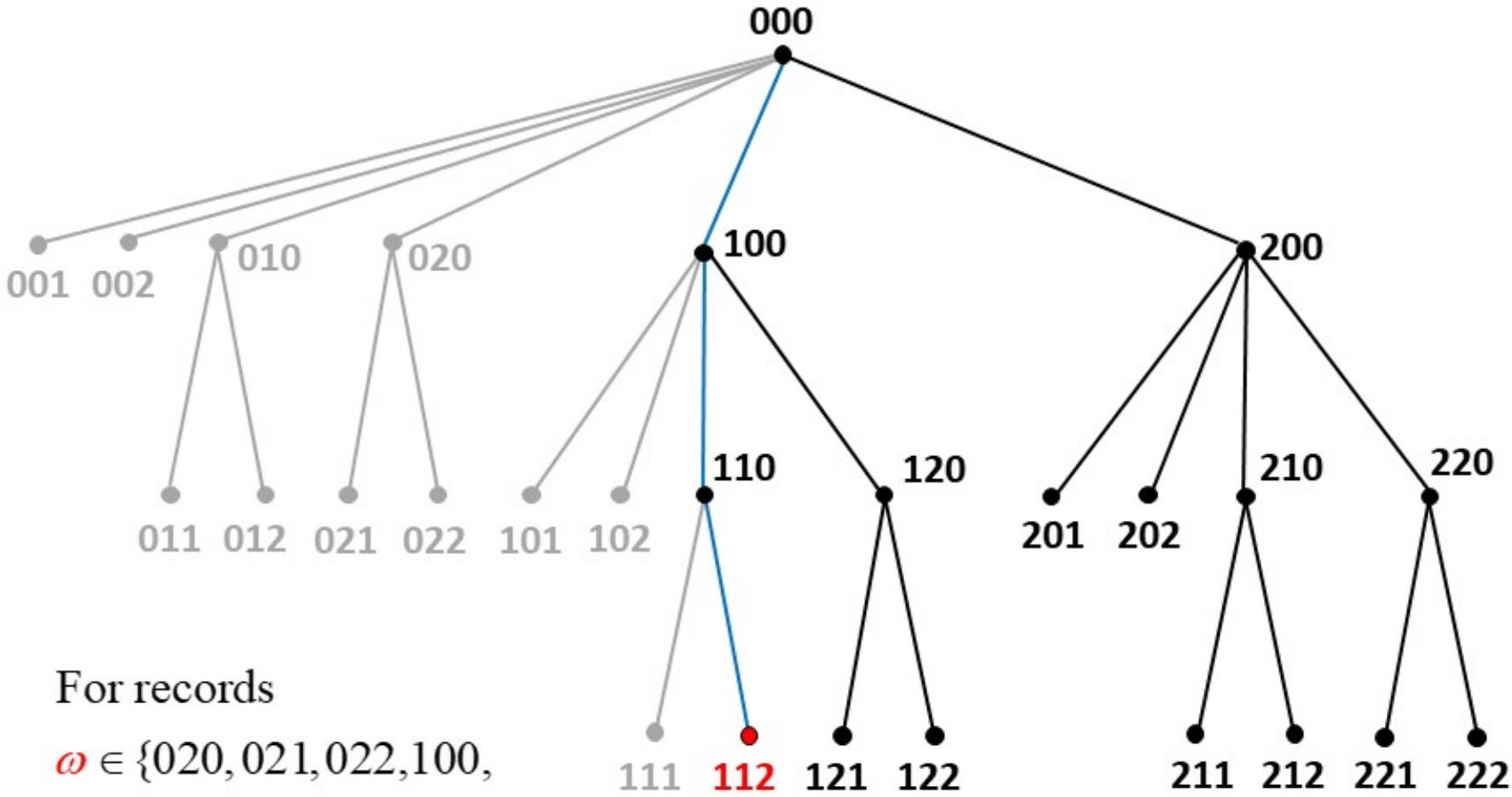
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 22. **Geometrical illustration of the program execution**



For records

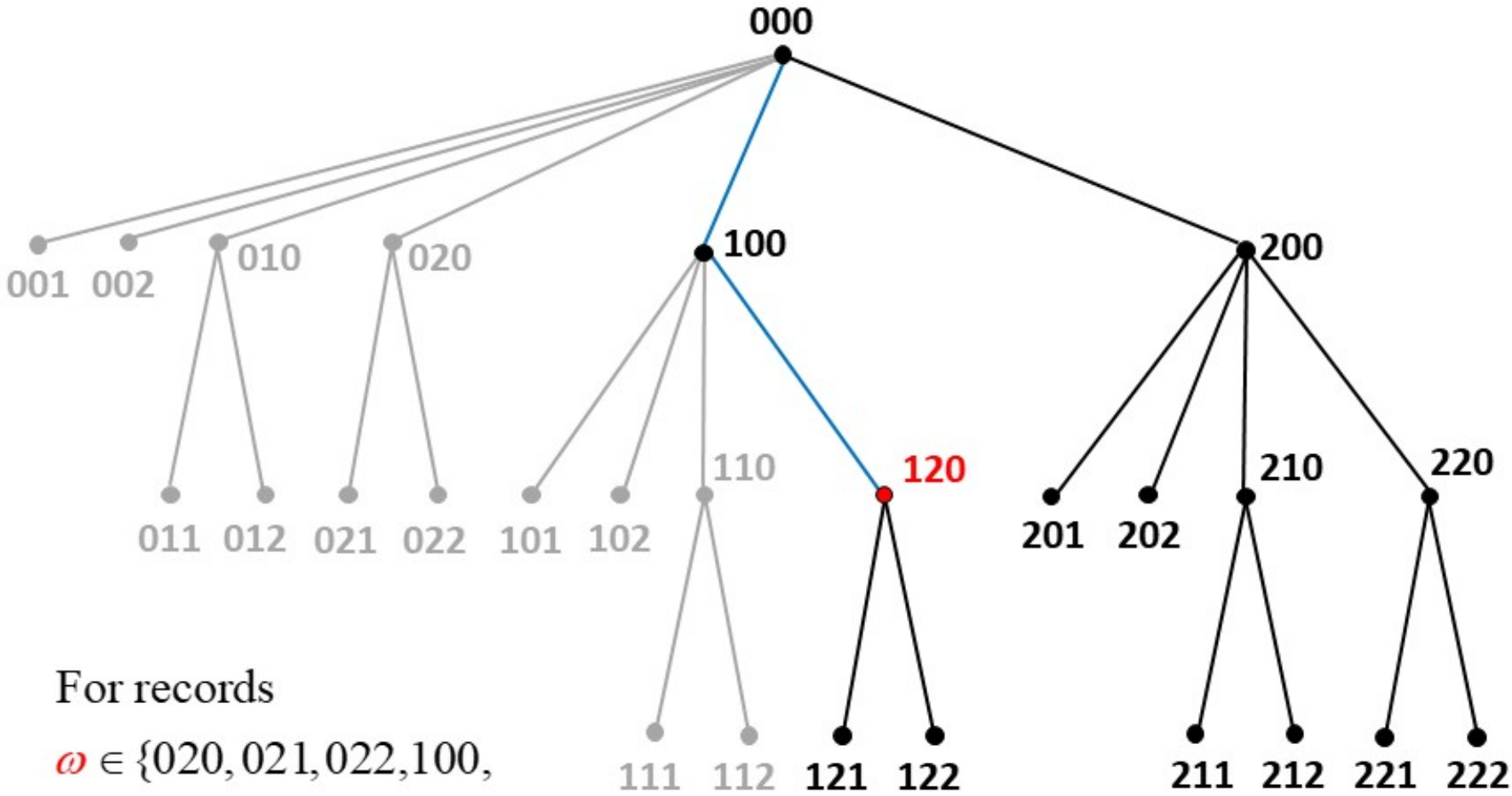
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 22. **Geometrical illustration of the program execution**



For records

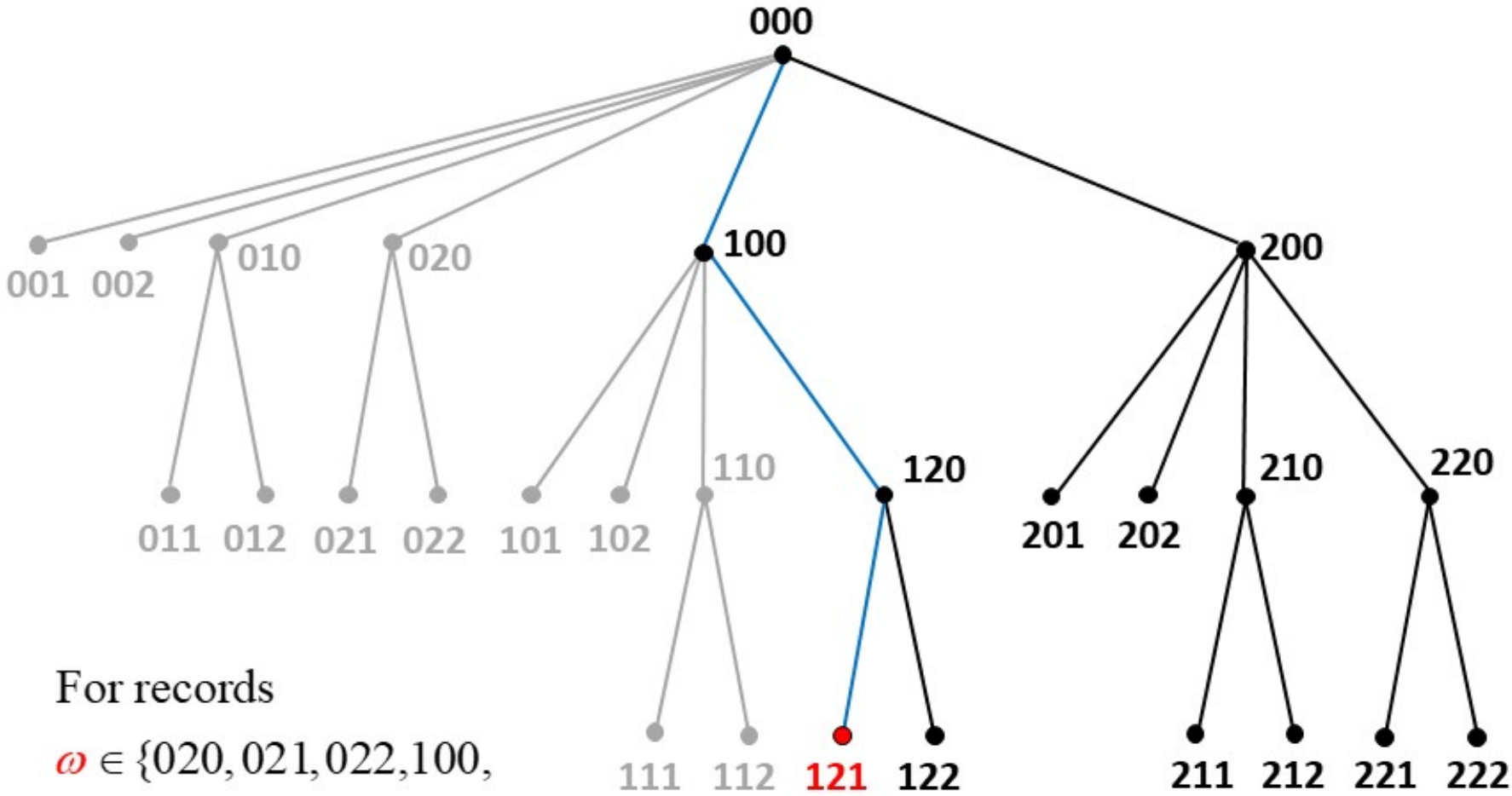
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 22. **Geometrical illustration of the program execution**



For records

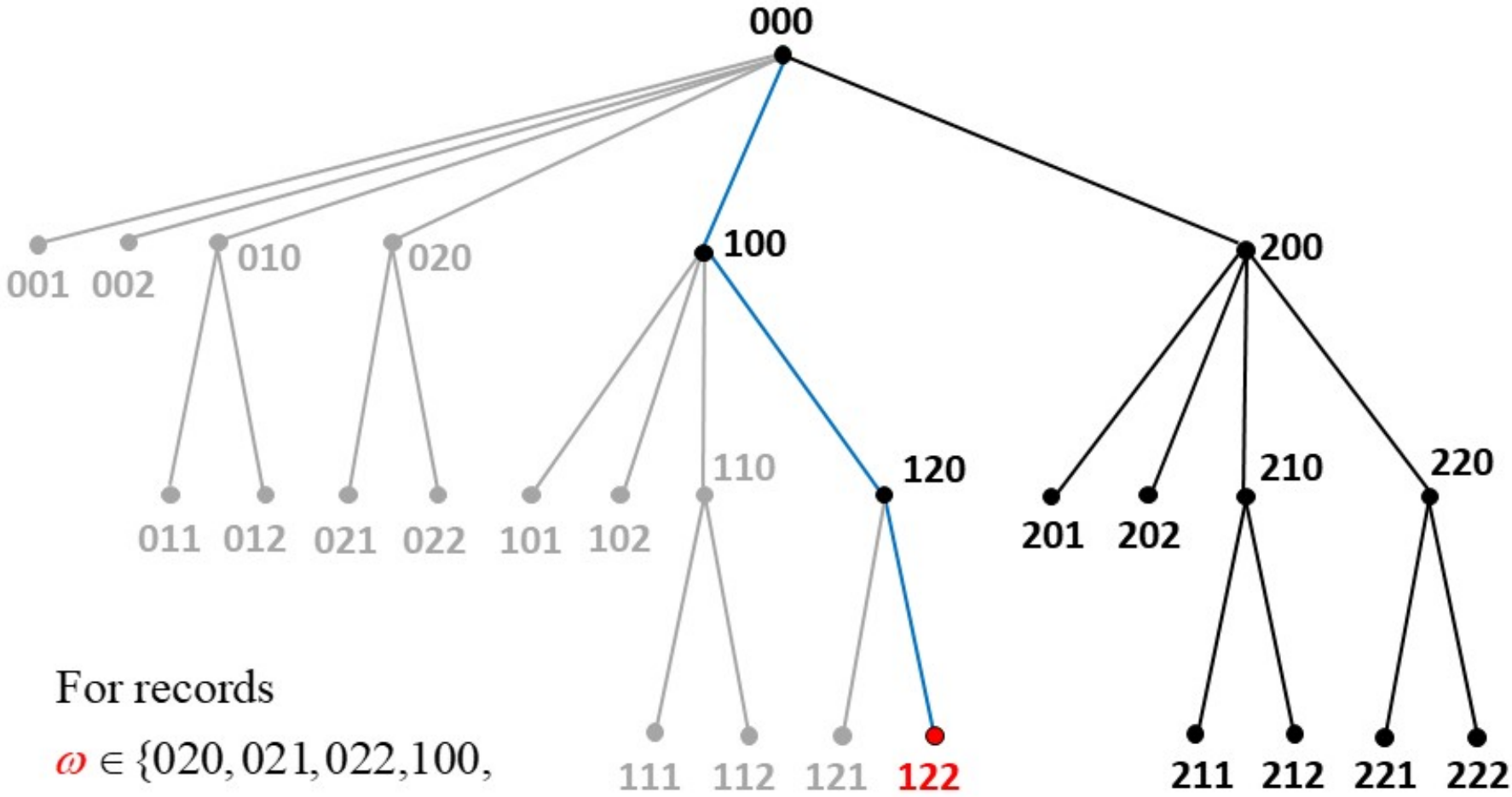
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 22. **Geometrical illustration of the program execution**



For records

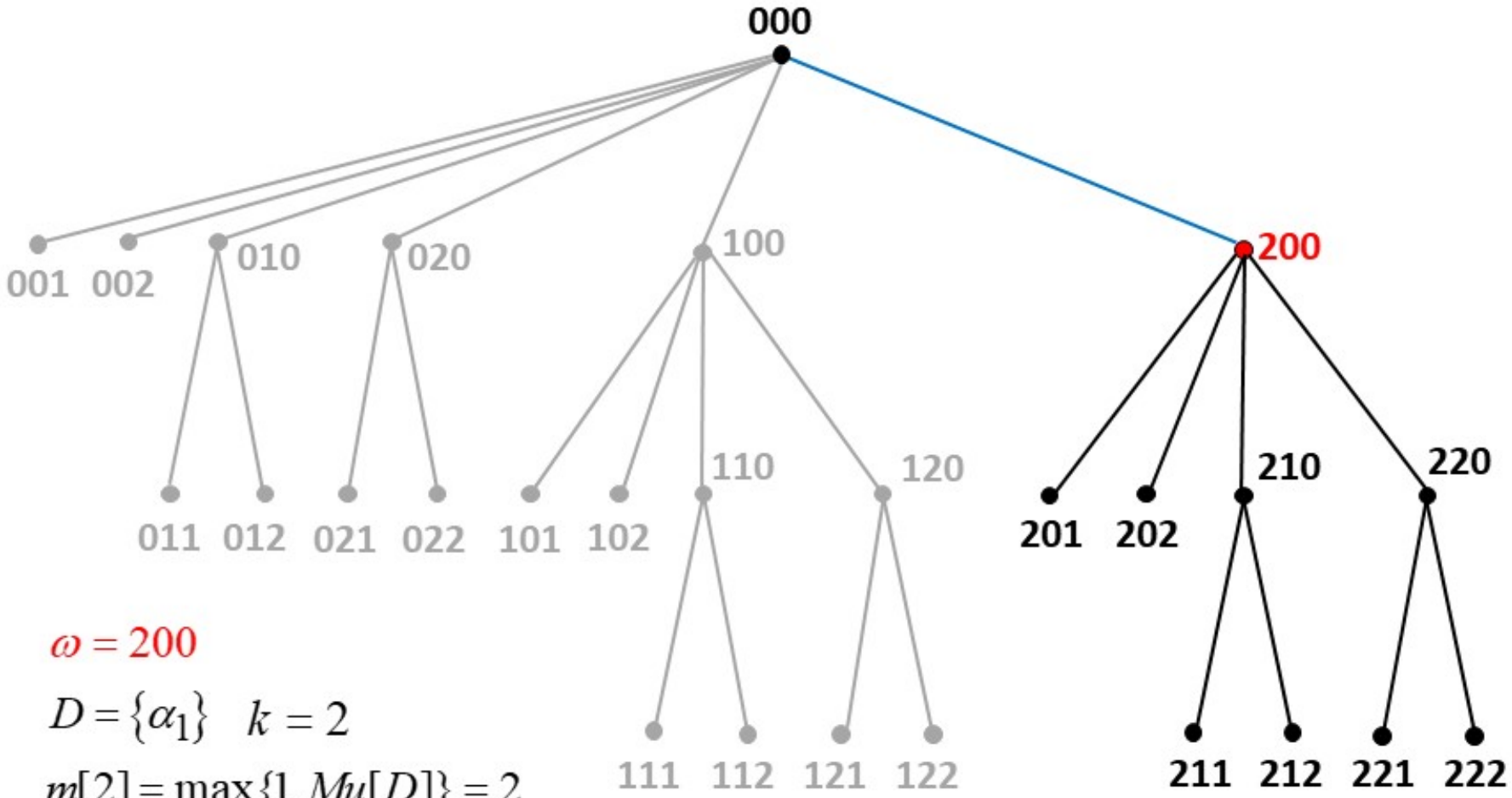
$\omega \in \{020, 021, 022, 100,$

$101, 102, 110, 111,$

$112, 120, 121, 122\}$

values $m[1], m[2], m[3]$ do not change.

Slide 23. **Geometrical illustration of the program execution**



$\omega = 200$

$D = \{\alpha_1\} \quad k = 2$

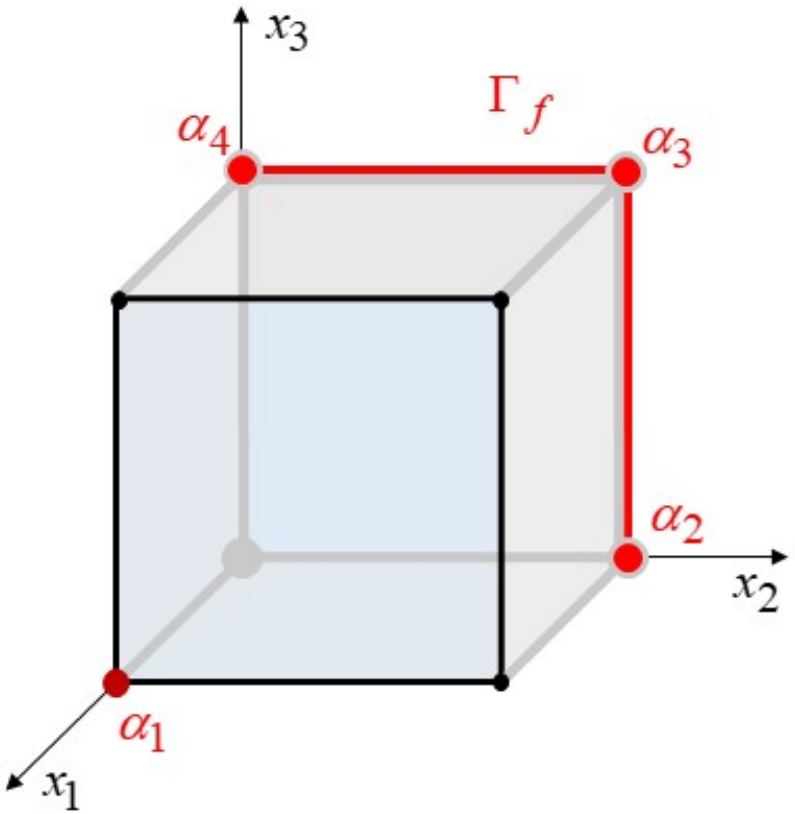
$m[2] = \max\{1, Mu[D]\} = 2$

$S \leftarrow \omega \quad C \leftarrow D$

$S = \{000, 200\} \quad C = \{C_f, \{\alpha_1\}\}$

$\omega = MinChild(\omega) = 201$

$\boldsymbol{m = (1, 2, 1)}$



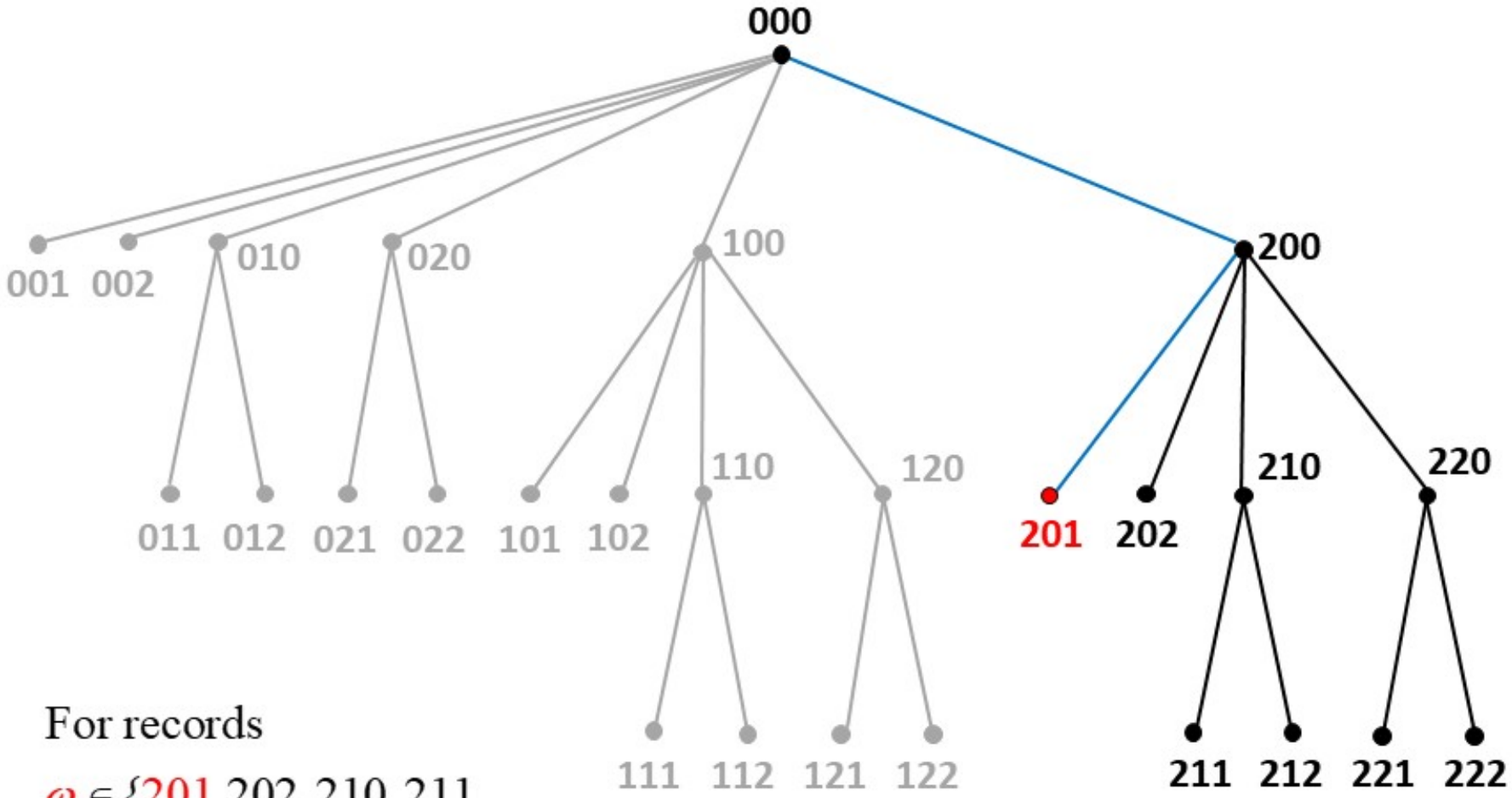
$\omega = 200$

$w = 1{*}{*} \quad x_1 = 1$

$D = \{\alpha_1\}$

$Mu(D) = 2$

Slide 23. **Geometrical illustration of the program execution**



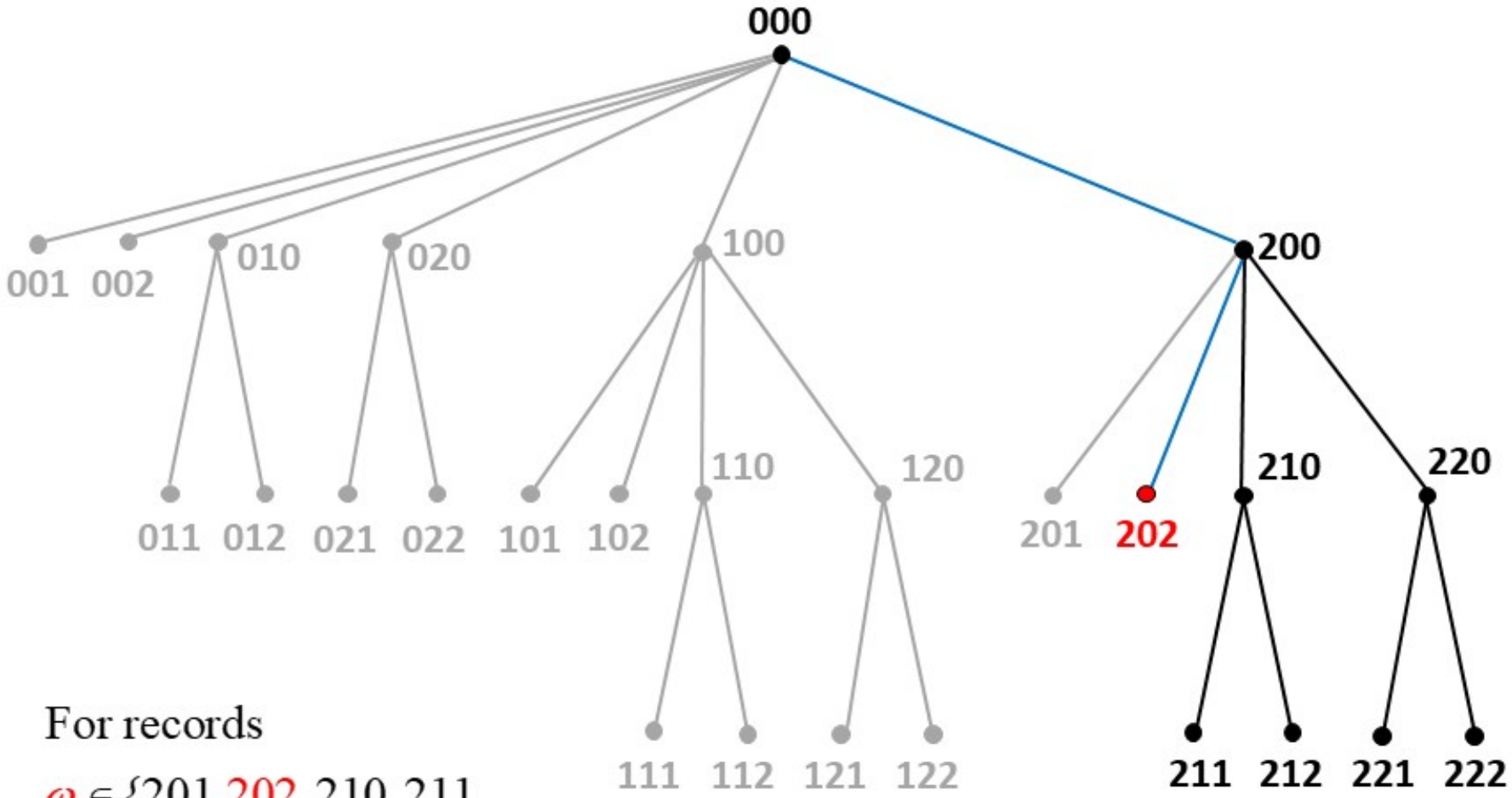
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 23. **Geometrical illustration of the program execution**



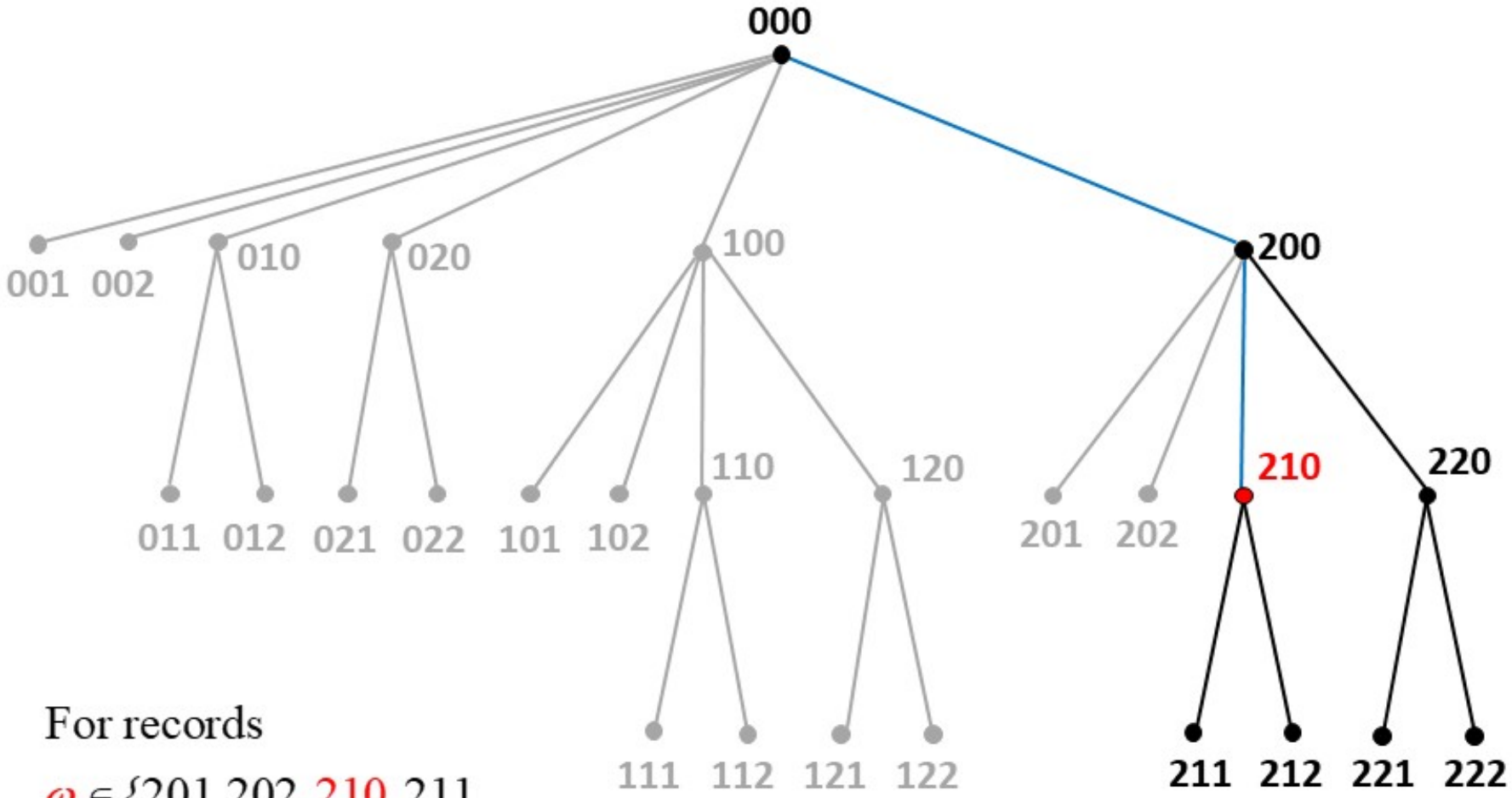
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 23. **Geometrical illustration of the program execution**



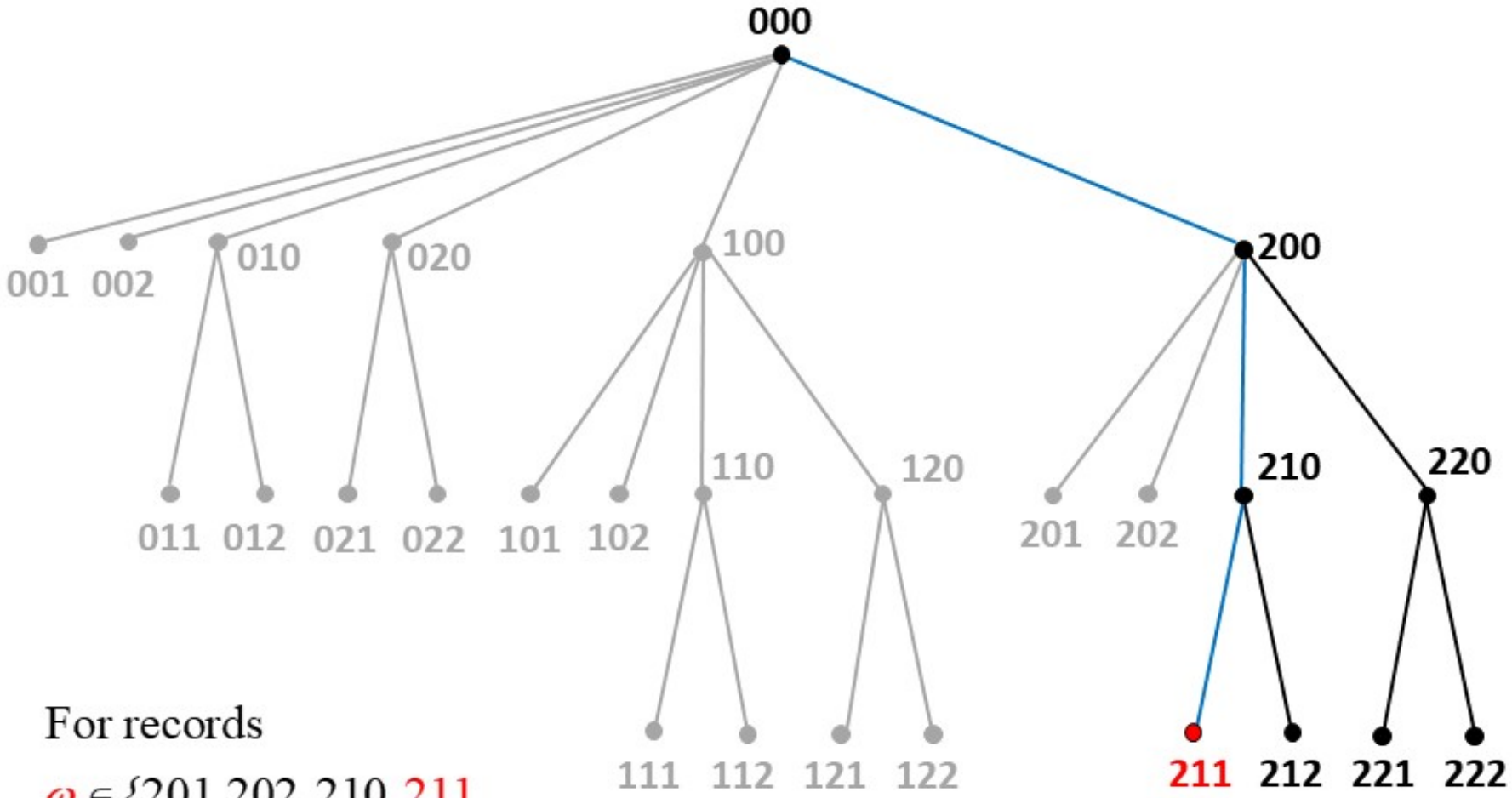
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 23. **Geometrical illustration of the program execution**



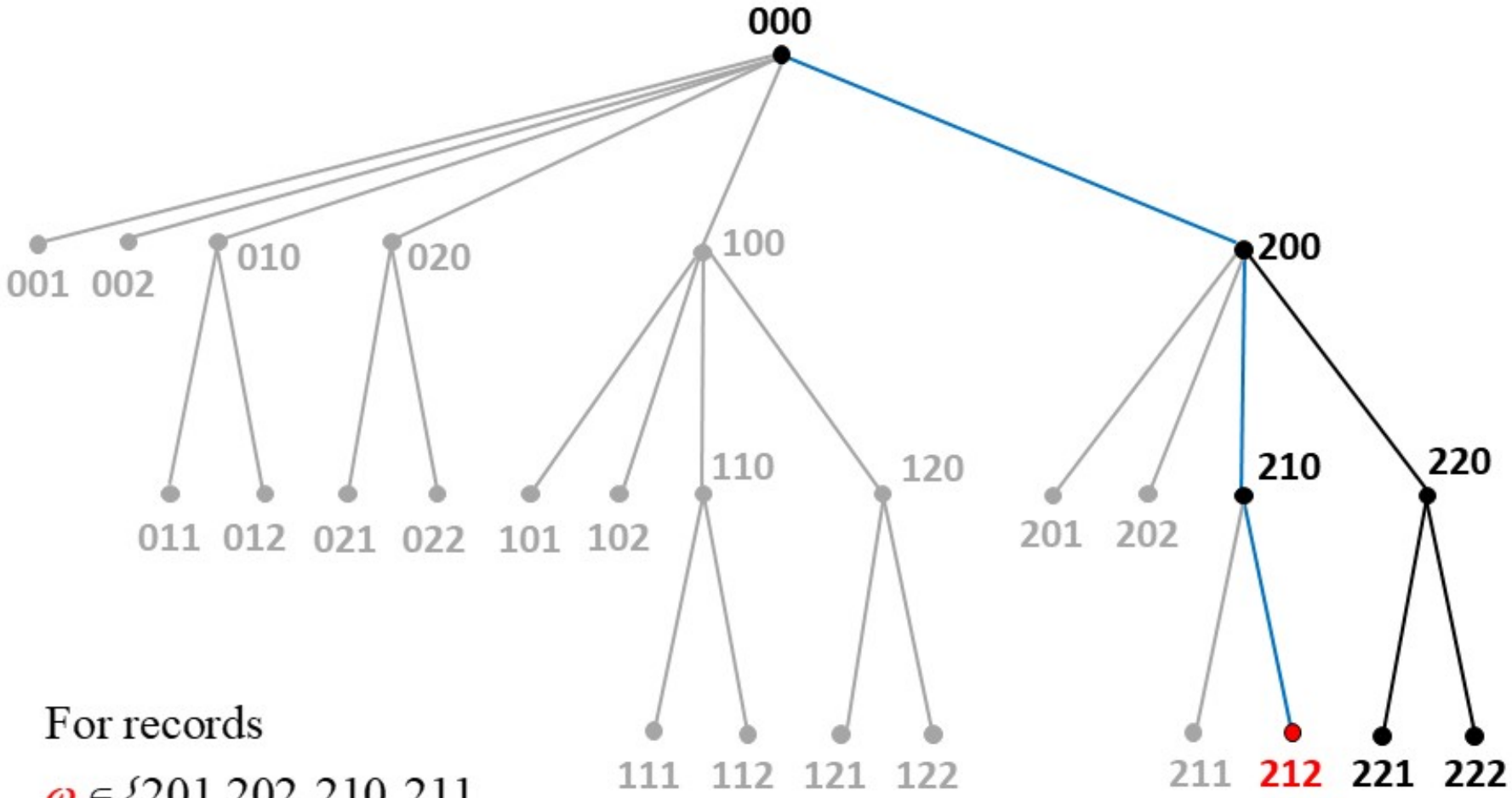
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 23. **Geometrical illustration of the program execution**



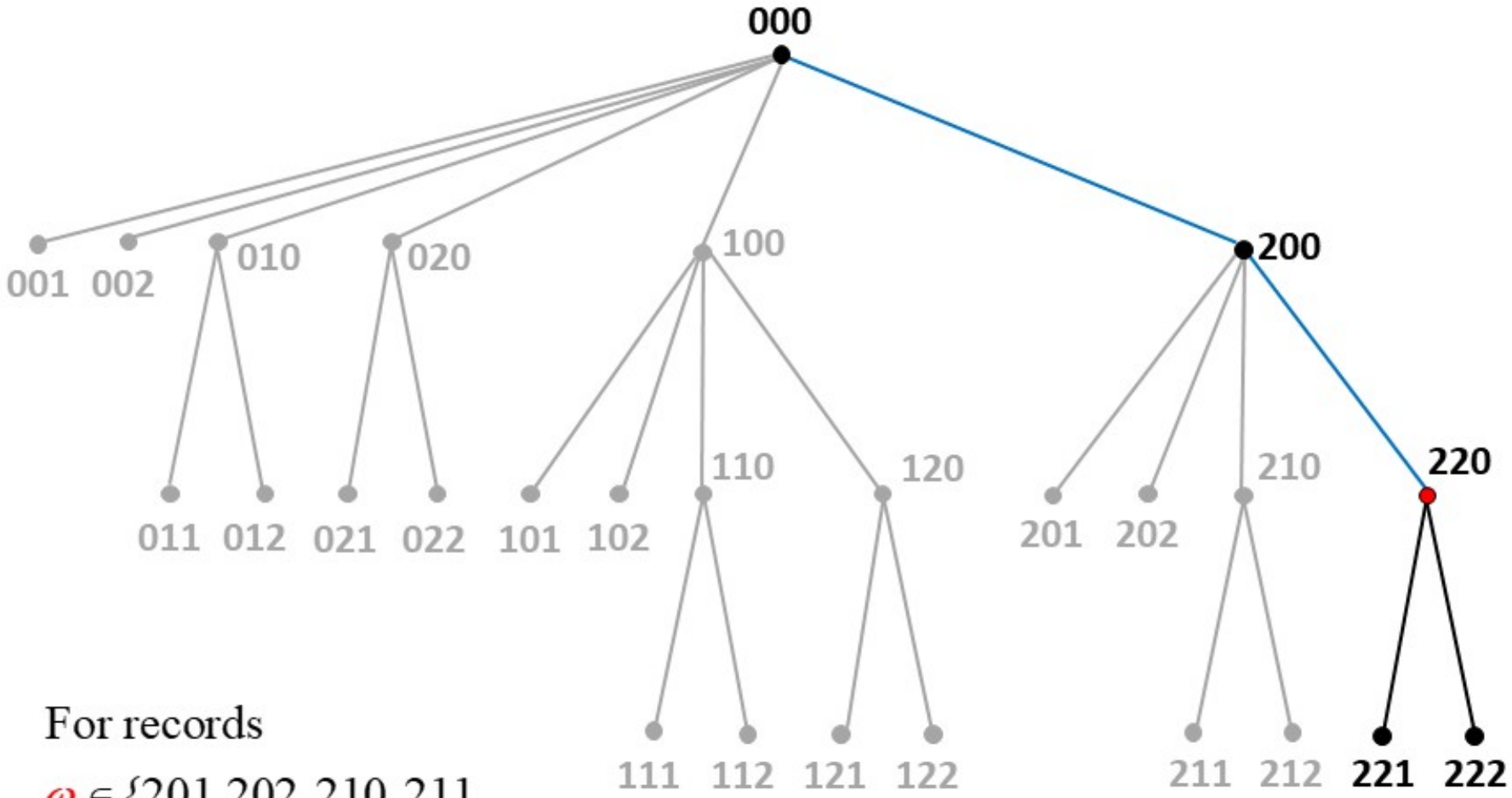
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 23. **Geometrical illustration of the program execution**



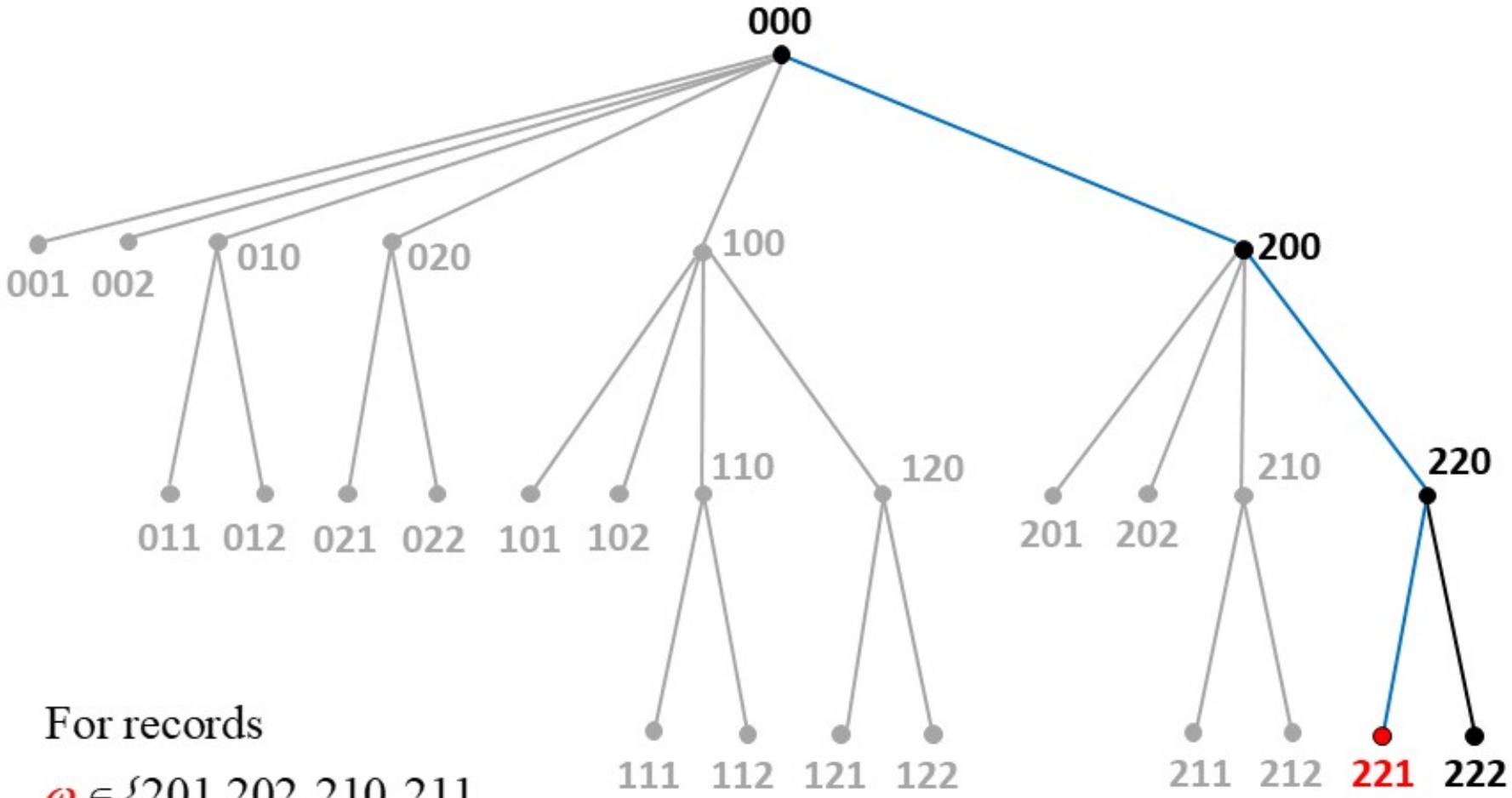
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 23. Geometrical illustration of the program execution



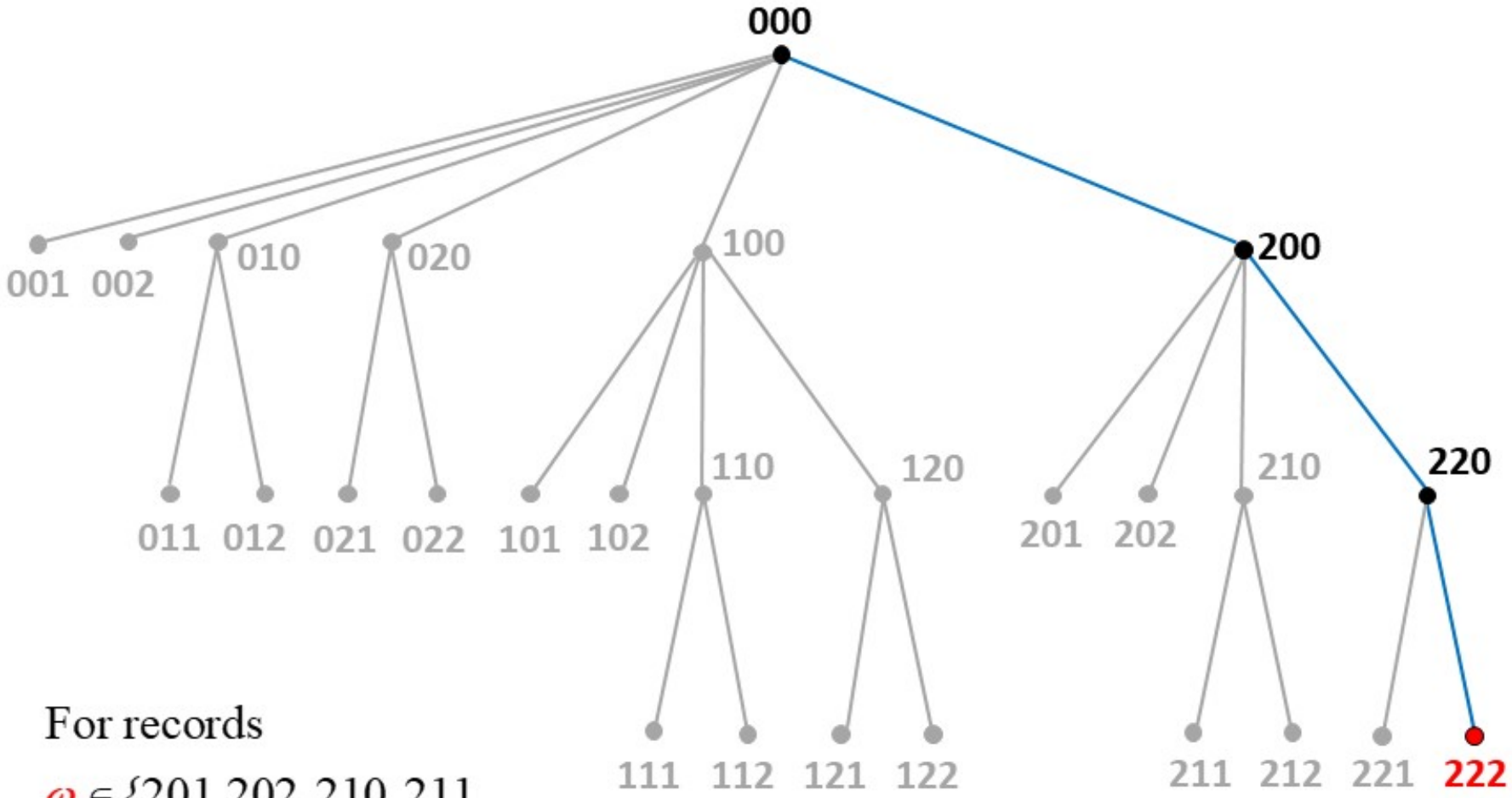
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 24. Geometrical illustration of the program execution



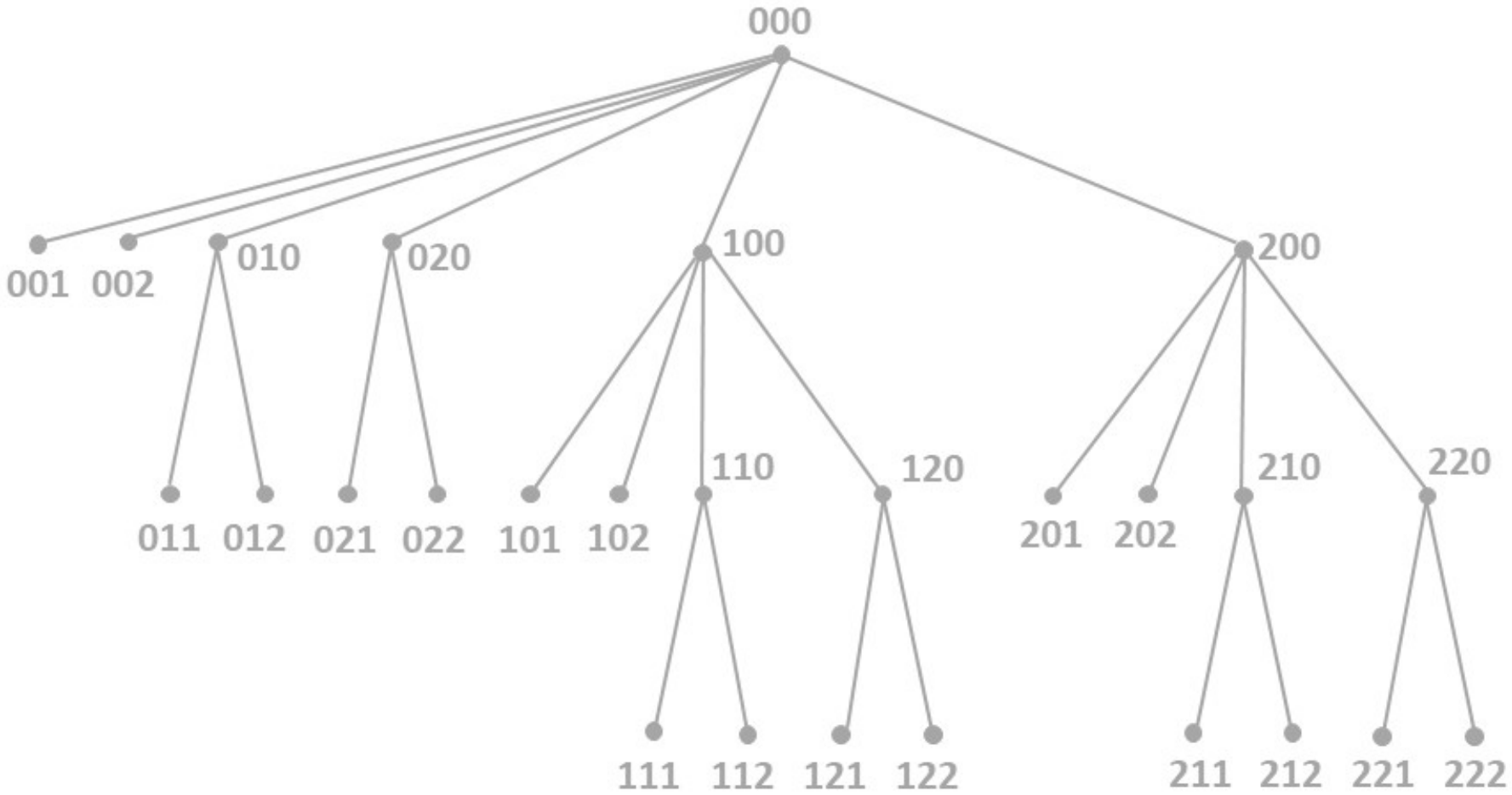
For records

$\omega \in \{201, 202, 210, 211,$

$212, 220, 221, 222\}$

values $m[1], m[2], m[3]$ do not change.

Slide 25. **Geometrical illustration of the program execution**



All the nodes are traversed

$$\boldsymbol{m} = (\mathbf{1}, \mathbf{2}, \mathbf{1})$$

1. K. Rothman, "Causes," *Am. J. Epidemiol.*, vol. 104, no. 6, pp. 587–592, 1976.

2. O. Miettinen, "Causal and preventive interdependence: Elementary principle," *Scand. J. Work. Environ. Health.*, vol. 8, pp. 159–168, 1982.

3. J. L. Mackie, "Causes and conditions," *Am. Philos. Q.* , vol. 2, no. 4, pp. 245–264, 1965.

4. T. J. VanderWeele and M. Robins, "A Theory of Sufficient Cause Interactions," *The Berkeley Electronic Press. COBRA Preprint Series*, p. Article 13, 2006, article 13.

5. J. L. Mackie,, *The cement of the Universe: a study of causation*. Clarendon Press, 1980.

6. D. Lewis, "Causation," *J. Philosophy*, vol. 70, pp. 556–567, 1973.

7. D. Lewis,, *Philosophical papers*. UK: Oxford University Press, 1987.

8. D. MacMahon and T. F. Pugh, *Preventive Medicine*. Boston: Little Brown, 1967, ch. Causes and entities of disease, pp. 11–18.

9. S. Greenland and C. Poole, "Invariants and noninvariants in the concept of interdependent effects," *Scand. J. Work Environ. Health.*, vol. 14, pp. 125–129, 1988.

10. T. J. VanderWeele and M. Robins, "The identification of synergism in the sufficient-component-cause framework," *Epidemiology*, vol. 18, no. 3, pp. 329–339, 2007.
11. T. J. VanderWeele and T. S. Richardson, "General theory for interactions in sufficient cause models with dichotomous exposures," *Ann. Statistics.*, vol. 40, pp. 2128–2161, 2012.
12. J. V. Nagrebetskaya and V. G. Panov, "Joint action of binary factors in the sufficient causes theory and its classification," *Int. J. Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 2146–2152, 2019.
13. J. Nagrebetskaya and V. Panov, "Spectrum of Joint Action of Factors in the Binary Theory of Sufficient Causes," in *2022 Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, 2022, pp. 208–211.
14. J. Nagrebetskaya, V. Panov, E. Vasilko, E. Vakhitov, and G. Ageev, "Effective Algorithms for Calculation of Joint Action Spectrum in the Theory of Sufficient Causes," in *2022 Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, 2022, pp. 212–215.

Thank you for your attention!